Proceedings

TWENTY-FIFTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

September 30-October 2, 1987

Allerton House, Monticello, Illinois
Sponsored by the
Coordinated Science Laboratory and the
Department of Electrical and Computer Engineering of the
University of Illinois at Urbana-Champaign

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

PHOTO REPRODUCTION

~~JUL 21 REC'D~~

~~SEP 29 REC'D~~

INTERLIBRARY LOAN

INTERLIBRARY LOAN

DUE: 12-14-88

DUE: 12/9/99

REC'D DEC 21 '88

DEC 16 REC'D

JUL 14 1989

JUN 26 REC'D

PHOTO REPRODUCTION

DEC 04 REC'D

SEP 5 REC'D

PHOTO REPRODUCTION

OCT 24 1990

L161—O-1096

PROCEEDINGS

TWENTY-FIFTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

VOLUME I

P. R. Kumar
Michael C. Loui
Conference Co-Chairmen

Conference held
September 30-October 2, 1987
Allerton House
Monticello, Illinois

Sponsored by
The Coordinated Science Laboratory
and
The Department of Electrical and Computer Engineering
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

Dedicated to

Mac E. Van Valkenburg

621.3811
Al5
1987:1

Engin.

FOREWORD

The Twenty-Fifth Annual Allerton Conference on Communication, Control, and Computing was held at Allerton House, Monticello, Illinois on September 30 through October 2, 1987. The conference was sponsored by the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. These *Proceedings* contain the papers presented at the Conference.

We are grateful to Professor W. M. Wonham of the University of Toronto for his plenary lecture "Logic and Language in Control Theory." We are also grateful to Professor B. H. Krogh of Carnegie-Mellon University for organizing the invited session "Discrete Event Systems" and to Professor M. W. Spong for organizing the invited session on "Robotics." We would like to thank all the session chairs and the authors for their contributions.

We appreciate the support of Professor M. E. Van Valkenburg, Dean of the College of Engineering and founder of the series of Allerton Conferences, Professor T. N. Trick, Head of the Department of Electrical and Computer Engineering, and Professor W. K. Jenkins, Director of the Coordinated Science Laboratory. We would also like to thank our colleagues Professors J. A. Abraham, E. Arikan, K. S. Arun, P. Banerjee, M. T. Basar, Y. Bresler, D. J. Brown, W. K. Fuchs, B. E. Hajek, T. S. Huang, R. Iyer, P. V. Kokotovic, H. Lee, J. V. Medanic, D. C. Munson, J. H. Patel, W. R. Perkins, K. R. Poolla, H. V. Poor, F. P. Preparata, M. B. Pursley, D. V. Sarwate, M. W. Spong, and B. W. Wah for their generous assistance in reviewing papers and organizing the papers into sessions.

We are grateful to Mr. Steven Williams for his handling of the business affairs of the Conference, to Mr. Robert Ebeling and his assistants for their timely printing of large numbers of programs and registration packets, and to Ms. Rose Harris, Ms. Tina Lipscomb, Ms. Dixie Murphy, and Ms. Janet Reese for their capable assistance in handling the Conference announcement, registration, program preparation, and editing and sales of these *Proceedings*.

P. R. Kumar

Michael C. Loui

TABLE OF CONTENTS

FOREWORD i

PLENARY LECTURE

LOGIC AND LANGUAGE IN CONTROL THEORY 1
W.M. Wonham
University of Toronto

I-A: COMBINATORIAL ALGORITHMS

Chairman: F.P. Preparata
University of Illinois at Urbana-Champaign

COMPLEXITY RESULTS FOR VERTEX-DELETION BIPARTIZATION AND VIA MINIMIZATION PROBLEMS 4
H-A. Choi, K. Nakajima, and C.S. Rim

ALGORITHMS FOR REGULAR PROPERTIES IN RECURSIVE GRAPHS 14
S. Mahajan and J.G. Peters

AN OPTIMUM $\theta(n \log n)$ ALGORITHM FOR FINDING A CANONICAL HAMILTONIAN PATH AND A CANONICAL HAMILTONIAN CIRCUIT IN A SET OF INTERVALS 24
G.K. Manacher and T.A. Mankus

COMPUTING A CONDITIONAL EDGE-CONNECTIVITY OF A GRAPH 34
A-H. Esfahanian and S.L. Hakimi

SMALL-LARGE DYNAMIC PROGRAMMING FOR TREES 36
J. McHugh and Y. Perl

APPROXIMATE STRING MATCHING WITH CONTEXT-DEPENDENCY 46
A.A. Bertossi, E. Lodi, F. Luccio, and L. Pagli

DETERMINISTIC MOVE-TO-REAR LIST ORGANIZING STRATEGIES WITH OPTIMAL AND EXPEDIENT PROPERTIES 54
B.J. Oommen, E.R. Hansen, and J.I. Munro

AN INVESTIGATION OF SELF-ORGANIZING HEURISTICS FOR DOUBLY LINKED LISTS 64
M.E. Bellow

HOW WELL ARE BINARY SEARCH TREES BALANCED? 66
W. Szpankowski and V. Rego

I-B: ROBUST AND NONLINEAR CONTROL

Chairman: R. Ortega
National University of Mexico

NECESSARY AND SUFFICIENT CONDITIONS FOR STABILITY OF POLYTOPES OF MATRICES THROUGH TESTS OF LOWER DIMENSIONAL SUBSETS 72
C.L. DeMarco

SLIDING REGIMES IN ANALOG SIGNAL ENCODING AND DELTA MODULATION CIRCUITS 78
H. Sira-Ramirez

FEEDBACK STABILIZATION OF UNCERTAIN SYSTEMS 88
P.P. Khargonekar, I.R. Petersen, and K. Zhou

OBSERVER-BASED DESIGN OF UNCERTAIN SYSTEMS: RECOVERING STATE FEEDBACK ROBUSTNESS UNDER MATCHING CONDITIONS 97
F. Esfandiari and H.K. Khalil

CONTROL OF A PM STEPPER MOTOR USING NONLINEAR OBSERVERS AND CONTROLLERS 107
M. Zribi and J. Chiasson

KHARITONOV'S THEOREM: IS THERE A GENERALIZATION FOR DEPENDENT COEFFICIENT VARIATIONS? 117
B.R. Barmish

EIGENVALUE BOUNDS FOR THE SOLUTION OF DISCRETE STOCHASTIC RICCATI EQUATION 119
E. Yaz

ROBUST FLIGHT CONTROL SYSTEM DESIGN TECHNIQUE USING GAIN SPACE WITH PERFORMANCE AND HANDLING QUALITIES CONSTRAINTS 121
G. Aziz and J. Ho

I-C: DETECTION

Chairman: W.L. Root
University of Michigan

SOME RESULTS ON THE ROLE OF 2-ALTERNATING UPPER PROBABILITIES IN SIGNAL-PLUS-NOISE MODELS 123
A. Sadrolhefazi and T.L. Fine

BOUNDS ON THE VARIANCES OF IMPORTANCE SAMPLING SIMULATIONS IN DIGITAL COMMUNICATION SYSTEMS ... 125
D. Lu and K. Yao

ON THE APPLICATION OF IMPORTANCE SAMPLING TO THE ANALYSIS OF DETECTION SYSTEMS ... 135
G. Orsak and B. Aazhang

WALD'S IDENTITY AND SEQUENTIAL TEST PERFORMANCE WITH DEPENDENT DATA ... 145
J.S. Sadowsky

DISTRIBUTED DETECTION WITH CORRELATED SENSOR NOISE ... 155
R. Viswanathan, V. Aalo, and S.C.A. Thomopoulos

A DIFFERENTIAL GEOMETRIC APPROACH TOWARD ROBUST SIGNAL DETECTION IN NONSTATIONARY AND/OR DEPENDENT NOISE ... 163
M.W. Thompson and D.R. Halverson

SIGNAL DETECTION IN ARCTIC UNDER-ICE NOISE ... 172
P.A. Nielsen and J. Thomas

I-D: DIGITAL SIGNAL PROCESSING

Chairman: K.S. Arun
University of Illinois at Urbana-Champaign

DETECTION OF SPEECH SIGNALS AT THE OUTPUT OF A COCHLEAR MODEL ... 182
B. Monderer and A.A. Lazar

LIMIT CYCLES IN LINEAR SHIFT VARYING DIGITAL FILTERS ... 192
T. Bose and D.P. Brown

BEST ORDERINGS FOR FLOATING POINT ADDITION ... 199
T.G. Robertazzi and S.C. Schwartz

A RATIONAL VIEW OF THE FAST FOURIER TRANSFORM ... 201
C.M. Fiduccia

SPLIT LEVINSON ALGORITHM FOR TOEPLITZ MATRICES OF ANY RANK PROOFILE ... 203
M.K. Ciliz and H. Krishna

AN ALGORITHM FOR ESTIMATING THE FREQUENCIES AND AMPLITUDE MATRICES CORRESPONDING TO MULTIDIMENSIONAL HARMONIC SIGNALS IN NOISE 205
C. Foias, A.E. Frazho, and P.J. Sherman

ON THE OPTIMUM DELAY IN FREQUENCY-SAMPLING ALE 215
S.M. Kuo and M.A. Rodriguez

SYNTHESIS AND DESIGN OF STRUCTURALLY PASSIVE FULLY RECURSIVE HALF-PLANE 2-D DIGITAL FILTERS 217
S. Basu and K. Mahtre

I-E: PARALLEL PROCESSING

Chairman: B. Wah
University of Illinois at Urbana-Champaign

CONCURRENT PROCESSING WITH RESULT SHARING: AN EFFICIENT COMPUTING PARADIGM 227
S. Krishnaprasad

PARALLELIZING 'LEAST-COST' BRANCH-AND-BOUND ALGORITHMS 236
G. Lee

THE SIMPLE SIMON PROGRAMMING ENVIRONMENT: A STATUS REPORT 238
J.E. Cuny, D.A. Bailey, J.W. Hagerman, and A.A. Hough

THE LATTICE STRUCTURE OF A BUS ORIENTED MULTIPROCESSOR MODEL 248
A.A. Lazar and T.G. Robertazzi

MIXED-MODE COMPUTING WITH THE PASM SYSTEM PROTOTYPE 258
S.A. Fineberg, T.L. Casavant, and T. Schwederski

ONE-TO-K COMMUNICATION IN DISTRIBUTED-MEMORY MULTIPROCESSORS 268
H-A. Choi, A-H. Esfahanian, and L.M. Ni

II-A: DISCRETE EVENT SYSTEMS

Chairman: B.H. Krogh
Carnegie Mellon University

ON SYNCHRONIC-DISTANCE AND B-FAIR RELATIONS FOR PETRI NETS 270
M. Silva and T. Murata

INFINITESIMAL PERTURBATION ANALYSIS OF A MULTICLASS ROUTING PROBLEM 279
P. Vakili and Y-C. Ho

AUGMENTED CHAIN ANALYSIS OF MARKOV AND SEMI-MARKOV PROCESSES 287
S.G. Strickland and C.G. Cassandras

WEAK CONVERGENCE OF SAMPLE PATH DERIVATIVES FOR THE WAITING TIME IN A SINGLE SERVER QUEUE 297
M.A. Zazanis

CONSTRUCTION OF A T-BASE AND DESIGN OF A PERIODIC FIRING SEQUENCE OF A LIVE AND BOUNDED PETRI NET 305
Q.W. Ge, T. Tanida, N. Ono, and K. Onaga

T-COMPLEMENT PETRI NETS 315
R.S. Sreenivas

CONTROLLED PETRI NETS AND MAXIMALLY PERMISSIVE FEEDBACK LOGIC 317
B.H. Krogh

SUPERVISORY CONTROL WITH INFINITE-STRING SPECIFICATIONS 327
J.G. Thistle and W.M. Wonham

DECIDABILITY FOR TEMPORAL LOGIC IN CONTROL THEORY 335
J.F. Knight and K.M. Passino

II-B: LAYOUT AND WIRING

Chairman: M. Sarrafzadeh
Northwestern University

A LINEAR TIME ALGORITHM FOR CIRCULAR PERMUTATION LAYOUT 345
C.S. Rim, N.J. Naclerio, S. Masuda, and K. Nakajima

LAYOUTS OF GRAPHS OF ARBITRARY DEGREE 355
D.D. Sherlekar and J. JaJa

A LP APPROACH FOR OPTIMAL RECTANGLE ROUTING 365
Y.Y. Sung

SINGLE JOG RIVER ROUTING 373
T-C. Tuan and S.L. Hakimi

CHANNEL ROUTING WITH SUPERTERMINALS 375
G. Gudmundsson and S. Ntafos

A SUBOPTIMAL SOLUTION TO THE VIA PLACEMENT PROBLEM 377
T.F. Gonzalez and S. Kurki-Gowdara

THREE-LAYER WIRABILITY OF PLANAR LAYOUTS 387
T.F. Gonzalez and S-Q. Zheng

ON FINDING OBSTRUCTION SETS AND POLYNOMIAL-TIME ALGORITHMS FOR GATE MATRIX LAYOUT 397
R.L. Bryant, M.R. Fellows, N.G. Kinnersley, and M.A. Langston

II-C: ESTIMATION AND FILTERING

Chairman: J.B. Thomas
Princeton University

OPTIMAL JOINT PARAMETER AND STATE ESTIMATION OF LINEAR STOCHASTIC MIMO SYSTEMS USING PSEUDO-LINEAR IDENTIFICATION (PLID) 399
M.A. Hopkins and H.F. VanLandingham

A HYBRID METHOD OF ESTIMATING THE PARAMETERS OF ARMA MODELS 409
S. Li, Y. Zhu, and B.W. Dickinson

RELATIVE SUFFICIENCY 417
R.F. Popoli and J.M. Mendel

OPTIMAL AND SUBOPTIMAL FILTERING FOR LINEAR SYSTEMS DRIVEN BY SELF-EXCITED POISSON PROCESSES 426
M.A. Ingram and A.H. Haddad

AN ESTIMATION PROBLEM WITH A RANDOMLY PERTURBED LINEAR OPERATOR 436
M. Mobed and W.L. Root

SOME COMMENTS ON ESTIMATION OF RANDOM VARIABLES 446
A.E. Wessel and G.L. Wise

AN ERROR ANALYSIS FOR THE KALMAN FILTER APPLIED TO DATA CORRUPTED BY MULTIPLICATIVE NOISE 453
B.S. Chow and W.P. Birkemeier

II-D: IMAGE PROCESSING I: ANALYSIS AND SYNTHESIS

Chairman: D.C. Munson, Jr.
University of Illinois at Urbana-Champaign

EFFICIENT RECOGNITION OF HANDPRINTED CHARACTERS FROM TABLET-BASED INPUT 463
D.D. Kerrick and A.C. Bovik

USE OF FRACTAL PARAMETERS TO DETERMINE LOCAL SPATIO-TEMPORAL CHARACTERISTICS OF A TURBULENT POLLUTANT FROM VIDEO IMAGES 473
A. Ait-Kheddache and S.A. Rajala

DETERMINATION OF VELOCITY FIELDS FROM TIME-SEQUENTIALLY SAMPLED DATA 483
M. Rahgozar and J.P. Allebach

FAST ITERATIVE IMAGE RESTORATION ALGORITHMS 493
A.K. Katsaggelos and S.N. Efstratiadis

RESTORATION OF IMAGES DISTORTED BY SAMPLING JITTER 503
P. Shi and R.K. Ward

VECTOR DITHERING: BINARIZATION THROUGH UNCONVENTIONAL IMAGE PLANE TESSELLATION 513
T.S. Rao and G.R. Arce

II-E: LINEAR SYSTEMS

Chairman: P. Albertos
Univ. Politecnica de Valencia, Spain

LINEAR STABLE UNITY-FEEDBACK SYSTEM: NECESSARY AND SUFFICIENT CONDITIONS FOR STABILITY UNDER NONLINEAR ADDITIVE PLANT PERTURBATIONS 522
C.A. Desoer and M.G. Kabuli

EXTENDED ZEROS AND MODEL MATCHING 524
M.K. Sain and B.F. Wyman

TRANSFER MATRICES, POLYNOMIAL FRACTIONS AND REALIZATION OF CONTINUOUS-TIME LINEAR TIME-VARYING SYSTEMS 534
E. Emre, H-M. Tai, and J.H. Seo

CONDITIONS FOR EXISTENCE OF AN OPTIMAL CONTROL IN THE LQ REGULATOR PROBLEM FOR SINGULAR SYSTEMS 544
J.D. Cobb

STABILITY PARAMETERIZATIONS AND HIDDEN MODES IN TWO DEGREES OF FREEDOM CONTROL DESIGN........ 546
P.J. Antsaklis and O.R. Gonzalez

ON THE DISTANCE FROM AN UNCONTROLLABLE PAIR: A SUMMARY........ 556
M.A. Wicks and R.A. DeCarlo

MINIMUM EFFORT STATE FEEDBACK POLE-PLACEMENT FOR LINEAR MULTIVARIABLE SYSTEMS........ 558
N.F. Al-Muthairi and S.P. Bingulac

MEASURES OF CONTROLLABILITY AND OBSERVABILITY AND RESIDUES........ 560
D.K. Lindner, J. Babendreier, and A.M.A. Hamdan

OBSERVER POLE ASSIGNMENT USING EIGENVALUE SENSITIVITY........ 562
M.A. Choudhry and C.S. Sims

III-A: COMMUNICATIONS AND CODING

Chairman: W.E. Stark
University of Michigan

WORST-CASE ANALYSIS OF SPREAD-SPECTRUM SYSTEMS WITH OUTPUT QUANTIZATION........ 564
M.V. Hedge, W.E. Stark, and D. Teneketzis

DYNAMIC CLIPPING FOR DETECTION OF FH/BFSK SIGNALS WITH DIVERSITY........ 574
Y.K. Kim and C.C. Lee

AN EXPURGATED LOWER BOUND ON THE FREE EUCLIDEAN DISTANCE OF TRELLIS CODES........ 584
M. Rouanne and D.J. Costello, Jr.

RANDOM CODING BOUNDS FOR A FAMILY OF BALANCED ERROR-CORRECTING CODES........ 594
M.A. Herro and V. Telang

ON GENERALIZED ERROR CORRECTING TRELLIS CODES WITH BALANCED BINARY SYMBOLS........ 596
H.C. Ferreira, D.A. Wright, and A.L. Nel

EFFICIENT MAXIMUM-LIKELIHOOD DECODING OF THE EXTENDED NORDSTROM-ROBINSON CODE........ 598
A.D. Abbaszadeh and C.K. Rushforth

SOFT DECISION DECODING OF CONCATENATED CODES 600
A.A. Hassan and W.E. Stark

EFFICIENT ALGORITHMS FOR THE SEARCH OF ZERO-CONCURRING CODEWORDS 610
A. Vardy and Y. Be'ery

CYCLIC CODES AND THEIR WEIGHTS 620
V. Pless

PERFORMANCE OF HYBRID ARQ CODING FORMATS OVER A BINARY SYMMETRIC CHANNEL 622
G.M. Huffman and F.D. Garber

III-B: PARALLEL AND DISTRIBUTED COMPUTATION

Chairman: G.N. Frederickson
Purdue University

A SIMPLE PARALLEL TREE CONTRACTION ALGORITHM 624
K. Abrahamson, N. Dadoun, D.G. Kirkpatrick, and T. Przytycka

AN EFFICIENT ASYNCHRONOUS DISTRIBUTED PROTOCOL TO TEST FEASIBILITY OF THE DUAL TRANSSHIPMENT PROBLEM 634
M.A. Comeau, K. Thulasiraman, and K.B. Lakshmanan

UNIVERSAL RANKINGS FOR BROADCASTING IN TREE NETWORKS 641
A. Rosenthal and P. Scheuermann

MINIMIZING SCHEDULE LENGTH SUBJECT TO MINIMUM FLOW TIME 650
J.Y-T. Leung and G.H. Young

AN OPTIMAL PARALLEL SOLUTION TO THE ECDF SEARCHING PROBLEM FOR HIGHER DIMENSIONS ON A MESH-OF-PROCESSORS 660
F. Dehne and I. Stojmenovic

SPORT-SORT SORTING ALGORITHMS AND SPORT TOURNAMENTS 662
H. Block

III-C: STOCHASTIC SYSTEMS

Chairman: R. Mazumdar
Columbia University

MODELING AND DYNAMIC SCHEDULING OF A QUEUEING SYSTEM WITH BLOCKING AND STARVATION 664
R.A. Milito and H. Levy

STRUCTURAL DECOMPOSITION OF MULTIPLE TIME SCALE MARKOV PROCESSES 674
J.R. Rohlicek and A.S. Willsky

THE SETTLING TIME OF A CLOSED JACKSON NETWORK 684
V. Anantharam

ON SCHEDULING TASKS WITH STOCHASTIC PROCESSING TIMES ON PARALLEL SERVERS 694
S.P.R. Kumar

GLOBAL CONVERGENCE OF RECURSIVE STOCHASTIC MINIMIZATION ALGORITHMS 700
S. Gelfand and S.K. Mitter

A GLOBAL OPTIMAL CONTROL SETTING FOR LARGE DEVIATIONS 707
E. Kappos

OPTIMAL CONTROL OF ITEMIZED PRODUCTION TO MEET UNCERTAIN DEMAND 713
J.B. Ribeiro do Val

STOCHASTIC APPROXIMATION FOR DISTRIBUTED PARAMETER SYSTEMS 721
N. Berman and A. Shwartz

III-D: IMAGE PROCESSING II: EDGE DETECTION AND RANKED-ORDER FILTERING

Chairman: T.S. Huang
University of Illinois at Urbana-Champaign

EDGE DETECTION BY COST MINIMIZATION 731
H.L. Tan, E.J. Delp, and S.B. Gelfand

REGULARIZED DIFFERENTIATION FOR EDGE DETECTION 741
A.K. Katsaggelos and B.J. Sullivan

EDGE ENHANCEMENT USING ORDER STATISTIC LAPLACIAN 743
Y.H. Lee and S.Y. Park

MULTILEVEL MEDIAN FILTERS 753
G.R. Arce and R.E. Foster

A FAST RECURSIVE TWO-DIMENSIONAL MEDIAN FILTERING ALGORITHM FOR MULTI-LEVEL IMAGE ROOT EXTRACTION 763
W.W. Boles, M. Kanefsky, and M. Simaan

CENSORED ORDER STATISTIC FILTERS 773
L. Naaman and A.C. Bovik

AN ANALYSIS OF A MULTIPLE MODEL MORPHOLOGICAL FILTER 775
J. Song and E.J. Delp

III-E: ADAPTIVE SYSTEMS

Chairman: J. Dorsey
Georgia Institute of Technology

DECENTRALIZED ADAPTIVE CONTROL OF LARGE SCALE SYSTEMS 785
H. Yousef and M. Simaan

ADAPTIVE CONTROL UNDER PARTIALLY MEASURED CONTROL VARIABLES 795
P. Albertos and F. Morant

DISTURBANCE-GENERATED BIFURCATION IN A PROTOTYPE ADAPTIVE SYSTEM WITH e_1-MODIFICATION LAW 801
S. Bai and F.M.A. Salam

ADAPTIVE CONTROL OF A CLASS OF DIFFERENTIAL DELAY SYSTEMS 811
F. Pourboghrat

SLIDING MODE PROPERTIES OF MIN-MAX CONTROLLERS AND OBSERVERS FOR UNCERTAIN SYSTEMS 813
S.H. Zak and R.A. DeCarlo

IV-A: ROBOTICS

Chairman: M.W. Spong
University of Illinois at Urbana-Champaign

ON THE STABILITY OF A MANIPULATOR INTERACTING WITH ITS ENVIRONMENT 821
J.E. Colgate and N. Hogan

CLOSED CHAIN MANIPULATOR DYNAMICS AND CONTROL 829
J.J. Murray and G.H. Lovell

PLANNING A COLLISION - AVOIDANCE PATH USING ANALYTIC MODELING FOR OBSTACLES 839
Y.M. Enab and J.Y.S. Luh

STABILIZATION OF UNCERTAIN SYSTEMS SUBJECT TO HARD BOUNDS ON CONTROL WITH APPLICATION TO A ROBOT MANIPULATOR 841
M. Hached, S.M. Madani-Esfahani, and S.H. Zak

FINDING EFFECTIVE 'FORCE TARGETS' FOR TWO-DIMENSIONAL, MULTIFINGER FRICTIONAL GRIPS 843
J.T. Schwartz and M. Sharir

IV-B: COMPUTATIONAL GEOMETRY

Chairman: H. Edelsbrunner
University of Illinois at Urbana-Champaign

A POLYGONAL APPROACH TO HIDDEN-LINE ELIMINATION 849
M.T. Goodrich

A DUAL APPROACH TO DETECT POLYHEDRAL INTERSECTIONS IN ARBITRARY DIMENSIONS 859
O. Gunther and E. Wong

FINDING LARGEST INSCRIBED EQUILATERAL TRIANGLES AND SQUARES 869
N.A. De Pano, Y. Ke, and J. O'Rourke

A NEW APPROACH TO THE DYNAMIC MAINTENANCE OF MAXIMAL POINTS IN A PLANE 879
G. Frederickson and S. Rodger

THE RECTILINEAR VORONOI DIAGRAM WITH BARRIERS 889
D.R. Fowler and J.M. Keil

SOME RESTRICTED RECTANGLE COVERING PROBLEMS 898
H.E. Conn and J. O'Rourke

AN EFFICIENT HEURISTIC ALGORITHM FOR A 3-SPACE n-PARTITIONING PROBLEM USED IN POLYGON RETRIEVAL AND NON-PARAMETRIC STATISTICS 908
P. Gburzynski and M. McLeish

IV-C: INFORMATION THEORY AND SOURCE CODING

Chairman: R.E. Blahut
IBM Corp. and University of Illinois at Urbana-Champaign

A COMPARISON OF CONDITIONS FOR POSITIVE CAPACITY OF AN ARBITRARILY VARYING CHANNEL 917
I. Csiszar and P. Narayan

MULTITERMINAL SOURCE ENCODING WITH ONE DISTORTION CRITERION 919
T. Berger and R. Yeung

HIGH-ORDER MARKOV ARITHMETIC CODING FOR DATA COMPRESSION 928
Y. Bar-Ness and S. Choi

ROBUST ADAPTIVE BUFFER-INSTRUMENTED ENTROPY-CODED QUANTIZATION OF STATIONARY SOURCES 938
J.W. Modestino, D.D. Harrison, and N. Farvardin

A PAIR OF INFORMATION THEORETIC LEMMAS WITH APPLICATION TO RUN-LENGTH CODING 946
C. Heegard

A COMPARISON OF THE WOLF-ZIV AND ALPHABET-CONSTRAINED APPROACHES FOR VECTOR QUANTIZATION OF NOISY SPEECH 955
J.D. Gibson, T.R. Fischer, and B. Koo

RATE-DISTORTION PERFORMANCE OF 2-D DPCM FOR GAUSSIAN AUTOREGRESSIVE SOURCES 965
J.W. Modestino and F. Azadegan

HIGHLY EFFICIENT ARITHMETIC CODING SCHEMES FOR LINE DRAWINGS BASED ON CHAIN CODE REPRESENTATIONS 967
J.G. Dunham and C-C. Lu

IV-D: COMMUNICATION NETWORKS

Chairman: K.S. Vastola
Rensselaer Polytechnic Institute

DECOMPOSITION OF A MULTIPLE ACCESS NETWORK INTO A NUMBER OF INTERCONNECTED SUBNETWORKS 969
M.A. Karatzoglu and A. Ephremides

DECENTRALIZED ALGORITHMS FOR OPTIMAL FLOW CONTROL 979
A.D. Bovopoulos and A.A. Lazar

ON PARETO OPTIMAL FLOW CONTROL IN A MULTICLASS ENVIRONMENT 989
C. Douligeris and R. Mazumdar

A (NOT VERY) SIMPLE ROUTING PROBLEM 998
P.P. Bhattacharya, I. Viniotis, and A. Ephremides

TERMINATION DETECTION IN A BROADCAST NETWORK 1007
T.H. Lai and T.H. Yang

ON THE PERFORMANCE OF SW-BANYAN NETWORKS 1017
M.H. Peyravi and M.R. Meybodi

THE COMPLEXITY OF ROUTING THROUGH AN OMEGA NETWORK 1027
P.J. Bernhard and D.J. Rosenkrantz

TWO-STAGE REALIZATIONS OF SYMMETRIC GROUPS 1037
A.Y. Oruc and S. Schneider

IV-E: TESTING AND FAULT TOLERANCE

Chairman: J.H. Patel
University of Illinois at Urbana-Champaign

TEST GENERATION FOR C-TESTABLE ARRAY MULTIPLIERS 1040
S-M. Chang and C-L. Wey

A FUNCTIONAL TEST SET FOR CMOS CIRCUITS 1050
G. Gupta and N.K. Jha

ON THE DESIGN OF PARALLEL SCAN TESTABLE FINITE STATE SEQUENTIAL MACHINES 1059
S. Iyengar, R. Dandapani, and S.M. Reddy

TRANSITION COUNT AS A COMPRESSION FUNCTION IN BUILT-IN SELF-TEST ENVIRONMENT 1069
K. Akiyama and K.K. Saluja

V-A: RANDOM ACCESS COMMUNICATIONS

Chairman: S. Verdu
Princeton University

MAXIMUM FINDING IN MULTIPLE ACCESS NETWORKS 1079
C.U. Martel and T.P. Vayda

COLLISION RESOLUTION ALGORITHMS FOR A TIME-CONSTRAINED MULTIACCESS CHANNEL 1081
S.S. Panwar, D. Towsley, J.K. Wolf, and Y. Armoni

ON DECENTRALIZED CONTROL ALGORITHMS FOR MULTIPACKET ALOHA 1089
S. Ghez, S. Verdu, and S.C. Schwartz

STABILITY CONDITIONS IN A MULTIQUEUE SYSTEM WITH APPLICATIONS TO THE ANALYSIS OF COMPUTER COMMUNICATION NETWORKS 1099
W. Szpankowski and V. Rego

THROUGHPUT ANALYSIS OF MAXEMCHUK'S INTEGRATED VOICE/DATA LOCAL AREA NETWORK PROTOCOL 1101
R. Kohli, A.C. Mingo, and K.S. Vastola

THE EFFECT OF EVENLY DISTRIBUTED STATIONS ON CARRIER SENSE MULTIPLE ACCESS PROTOCOL 1103
J.M. Vanderwiele and J.G. Bredeson

V-B: NEURAL NETWORKS

Chairman: V. Anantharam
Cornell University

QUALITATIVE ANALYSIS OF NEURAL NETWORKS: GLOBAL THEORY 1113
J-H. Li, A.N. Michel, and W. Porod

QUALITATIVE ANALYSIS OF NEURAL NETWORKS 1123
A.N. Michel, J.A. Farrell, and W. Porod

CONVERGENCE ISSUES IN PERCEPTRON BASED ADAPTIVE NEURAL NETWORK MODELS 1133
Y. Shrivastava and S. Dasgupta

ON THE DESIGN OF NEURAL NETS 1142
F.M.A. Salam, M.R. Choi, Y. Wang, and B. Javidi

V-C: STABILITY THEORY AND OPTIMAL CONTROL

Chairman: D.K. Lindner
Virginia Polytechnic Institute

A NEW INCLUSION CONCEPT AND OPTIMAL CONTROL FOR SYSTEMS WITH OVERLAPPING INFORMATION SETS 1152
A. Iftar and U. Ozguner

DISTRIBUTED COMPUTATIONS OF OPTIMAL CONTROL STRATEGIES FOR A CLASS OF INTERCONNECTED SYSTEMS 1162
L. Acar and U. Ozguner

A STABILITY THEORY OF LINEAR TIME VARYING SYSTEMS 1172
P.R. Bouthellier and B.K. Ghosh

FEEDBACK SYSTEMS WITH TAPERED AND OTHER SPECIAL INPUT SPACES 1181
W.L. Root and D. Serakos

PARALLEL GRADIENT PROJECTION ALGORITHMS TO SOLVE THE DISCRETE LQR OPTIMAL CONTROL PROBLEM WITH HARD CONTROL BOUNDS 1190
G.G.L. Meyer and L.J. Podrazik

CONJUGATE POINTS IN OPTIMAL CONTROL 1200
J. Kogan

TIME SENSITIVITY ANALYSIS OF NONCONVENTIONAL SAMPLED-DATA SYSTEMS 1202
P. Albertos and J. Tornero

V-D: FOUNDATIONS OF COMPUTING

Chairman: D.E. Muller
University of Illinois at Urbana-Champaign

INTRODUCTION TO DIALECTICAL NETS 1204
R.E. Kent

ON EXACT, CRITICAL AND UNIQUE VERSIONS OF SOME GRAPH PROBLEMS 1214
L.E. Rosier and H-C. Yen

COMPLEXITY AND DECIDABILITY FOR RESTRICTED CLASSES OF PICTURE LANGUAGES 1224
C. Kim

STRUCTURAL PROPERTIES OF COMMUNICATION STRUCTURES GENERATED BY AGGREGATE REWRITING GRAPH GRAMMARS 1234
D.A. Bailey and J.E. Cuny

INDEX OF AUTHORS

LOGIC AND LANGUAGE IN CONTROL THEORY

W.M. WONHAM
Systems Control Group
Dept. of Electrical Engineering
University of Toronto
Toronto, Ontario, Canada M5S 1A4

SUMMARY

In this talk we review the historical steps through which formal logic and formal language have entered control theory, and survey some of the current trends in control research in which logic and language provide the main methods.

While it is only recently that logic and language have begun to play a formal role in control theory, the origins of logical decision-making and "rule-based" structures in control can be found in the decade 1948-57, an era that might be called The Heroic Age of Cybernetics. While the decade was dominated by the ideas in the celebrated monographs of Norbert Wiener there appeared, in addition, some quite different "cybernetic" concepts, of which two of the most notable were Ross Ashby's *homeostat* (1948) and the *on-off* or *bang-bang* control of Donald Bushaw (1953).

The homeostat was designed to show that quasi-biological behavior involving adaptivity and homeostasis could be exhibited by a "machine" operating in accordance with the following simple control rule: If the state vector tries to cross the boundary of its admissible set, randomly select a new tuple of dynamic parameters; keep on doing this until the admissible set is made dynamically invariant. Ashby proposed the term *ultrastability* for the stability achieved by this kind of logic-based (in current jargon, "rule-based") decision-making. The idea was greeted by H.S. Tsien (1954) as a possible approach to fault-correction in aircraft controllers, although the method of random parametric selection was much too inefficient to lead to practical self-organizing systems.

More influential were the investigations of time-optimal relay (bang-bang) control pioneered by Flugge-Lotz and pursued in depth by Bushaw. Bang-bang systems are also rule-based: When the state vector hits the switching surface, reverse the polarity of control; keep on doing this until you reach the origin. The rigorous generalization of this idea to a broad class of nonlinear systems constituted the first great success of Pontryagin's Maximum Principle and (in control theory) of Bellman's Dynamic Programming, which combined to inaugurate the new decade starting in 1958.

These breakthroughs were quickly followed by a third: the Kalman-Bucy filter (1960), and Kalman's paradigm of a state-based linear model in which the new concepts of controllability and observability were fundamental. (Controllability had originated in the nonlinear bang-bang theory; observability was crucial in Kalman-Bucy filtering, and in this linear setting was seen to be algebraically dual to controllability.)

All this (plus space-program funding) generated the tremendous confidence and creativity of the '60s that resulted in what was called (in the '70s) Modern Control Theory. Control theory (no longer known in English as "cybernetics" because of the unprofessional connotations which that fine word somehow acquired for the Western engineering community) seemed now to be well established as an engineering science. Having mastered the optimal guidance of space vehicles, surely control theorists could now solve the control problems of Large Systems (traffic, industrial, organizational ...) as well!

Unfortunately, things turned out otherwise. Control theory was too specialized to differential models, and seemed unsupportive of *modularity*, that crucial ingredient of any decent large-scale system theory. On the asserted grounds that complex systems defy modeling by standard mathematics, Lotfi Zadeh proposed (1965) *Fuzzy Culture*. A typical control rule might be: "To sweeten the coffee, stir in some sugar; when the coffee is sweet enough, drink it!" In other words, express the control via "linguistic variables" (to capture the intuition of an experienced practitioner), relegating the details of how those variables are numerically evaluated to lower-level ("fuzzy") interpolation and decision procedures. Thus you may obtain a natural and implementable control decomposition. This semi-heuristic approach to the control of poorly-modeled systems (e.g. cement kilns...) was taken further by Mamdani and others in the '70s and is still influential. The point, for us, is that with Fuzzy Culture linguistic considerations, albeit in a rudimentary sense, had entered control theory for the first time.

As one other voice crying in the wilderness, we recall that of Yu. I. Klykov and D.A. Pospelov with their *situational control of large systems* (1970). In today's terminology the idea was to construct an expert system based on certain linguistic principles involving semantic networks and a theory of *generalization*, or partitioning of the class of *situations*. The objective was to match a combinatorially enormous (e.g. 2^{100}) state set to a reasonably small number (e.g. 1000) of possible control decisions. While the published details are sketchy, the approach was claimed to be highly successful in designing computer-assisted control of a seaport (ships, piers, cranes,...), achieving a saving of 600,000 rubles per year by reduction of "idle time" of ships, workers,... . Whether or not the workers were grateful was not reported. In any event, the formalism was logic- and language-based, thus articulating the unconventional view that a general and powerful control theory might be based on such concepts.

So much for developments in control to about 1975. Meanwhile, in computer programming circles, a great deal of effort was going into the development of logics for the verification of program correctness. Two of the major milestones were Hoare's (1969) axiomatic approach to computer programming and Dijkstra's (1976) view of a program (or program fragment) as a *predicate transformer*. Given a program PROG and a desired postcondition (on termination) PRED_out, one calculates (in principle) the *weakest precondition* WPRED_in on the input data that guarantees that PRED_out will be true. If the input data are actually known to satisfy some condition PRED_in, then PROG is *correct* just when PRED_in implies WPRED_in.

One way to relate this paradigm to control ideas is the following. Consider a *discrete-event system*, namely a system that's discrete in time and state-space, and typically asynchronous and nondeterministic. Model it by a state-transition structure in which certain transitions, the *controllable events*, can be enabled or disabled, similar to the action of a green or red light at a traffic intersection. Using state feedback one can, as usual, synthesize a family of dynamic flows. Now suppose that you want to make some predicate PRED on the state set dynamically invariant: for instance, if you start in an admissible ("safe") region of state space then under the action of control you would like to remain there (cf. the homeostat!). It's easy to see that invariance can be achieved with suitable feedback just when PRED is already invariant on the events that are *uncontrollable*, in which case we say that PRED is *control-invariant*. This is exactly the requirement that PRED implies its own weakest precondition for each uncontrollable event! If the given predicate PRED is not control-invariant, then it can be optimally approximated by a stronger (more restrictive) predicate PRED* that is. So if the system starts with PRED* true, then PRED* is guaranteed to remain true, and therefore so is PRED, and this is the best we can do.

More recently, programming theorists have proposed *temporal logic* for the verification of concurrent program correctness (cf. Hailpern, Manna & Pnueli and others, starting about 1982). Temporal logic is an extension of first-order (predicate) logic in which the values of variables, and the truth values of predicates, can change with time as the process of interest evolves from one state to the next. Again, reading "system" for "process" and bringing in feedback, one may expect a natural blending of logic with control ideas. In fact, *temporal operators* appearing as formal constructs in temporal logic can be used to express various standard control system requirements in a succinct way. Two standard operators are the *henceforth* operator $\Box$ and the *eventuality* operator $\Diamond$. Thus $\Box$PRED means "PRED is true from now on", while $\Diamond$PRED means "PRED will be true some time in the future". Sometimes $\Box$PRED is called a *safety* property, while $\Diamond$PRED is called a *liveness* property. Now in control theory, *regulation* means that the system should eventually enter and remain within a specified subset of its state set. Let this target subset be the truth set of a state predicate PRED_target, and let the initial state(s) form the truth set of PRED_init. Then the regulation property will hold exactly when PRED_init implies $\Diamond\Box$PRED_target. In any particular context the regulation problem will be solvable exactly when some feedback control can be synthesized such that this implication holds true.

To formally prove a temporal logic proposition requires a rather elaborate development of axioms and rules of inference. An exercise of this sort applied to a problem of regulation in a simple distributed system (originally proposed by Dijkstra) has been presented by Thistle and Wonham (1986). But the main point now is that, with temporal logic, we have moved much closer to a formal logic that matches the dynamic requirements of control theory, and the development in this setting of synthesis procedures for discrete control systems is an exciting current possibility.

Starting with the thesis of P.J. Ramadge (1983) a control theory has evolved for discrete-event systems that is purely language-based. The idea is to regard a DES as the generator of a formal language, and to define as *controllable* those sublanguages that can be synthesized by a closed-loop structure in which controllable events are disabled in accordance with some rule based on the history of the plant output. Such a formulation is fairly natural for a wide class of coordination problems in distributed controlled systems involving several initially independent subsystems. It turns out that any given language declared as a specification of "legal" behavior can be optimally approximated by a controllable language; namely the problem of synthesizing legal behavior has a unique optimal solution. The theory also includes a concept of *observable language*, and supports modular control in decentralized and hierarchical architectures.

In his 1987 thesis, J. Ostroff proposed using temporal logic as a tool for both system specification and verification, in the *real-time* control of *extended state machines*. And it seems reasonable to suppose that the language-based control theory of discrete-event systems just described can be meshed with temporal logic as well.

In conclusion, we see control science as the systematic study of how information and dynamics may be brought into purposeful interaction. It is defined in terms of problems and concepts, not in terms of techniques. Stimulated by the demands of technology and by developments in computer science, control science has entered a new phase, where discreteness, modularity, and communication among subsystems are fundamental. In this new phase, logic and language will be essential tools of thought.

COMPLEXITY RESULTS FOR VERTEX-DELETION BIPARTIZATION AND VIA MINIMIZATION PROBLEMS

HYEONG-AH CHOI [1]
Department of Computer Science
Michigan State University, East Lansing, MI 48824
KAZUO NAKAJIMA [2]
Electrical Engineering Department, Institute for Advanced Computer Studies,
and Systems Research Center
University of Maryland, College Park, MD 20742
CHONG S. RIM [2]
Electrical Engineering Department, and Systems Research Center
University of Maryland, College Park, MD 20742

ABSTRACT

The vertex- (resp., edge-) deletion graph bipartization problem is the problem of deleting a set of vertices (resp., edges) from a graph so as to make the remaining graph bipartite. In this paper, we first show that the vertex-deletion graph bipartization problem has a solution of size k or less if and only if the edge-deletion graph bipartization problem has a solution of size k or less, when the maximum vertex degree is limited to three. This immediately implies that (1) the vertex-deletion graph bipartization problem is NP-complete for cubic graphs, and (2) the minimum vertex-deletion graph bipartization problem is solvable in polynomial time for planar graphs when the maximum vertex degree is limited to three. We then prove that the vertex-deletion graph bipartization problem is NP-complete for planar graphs when the maximum vertex degree exceeds three. Using this result, we finally show that the via minimization problem, which arises in the design of integrated circuits and printed circuit boards, is NP-complete even when the maximum "junction" degree is limited to four.

1. INTRODUCTION

A *graph* G is an ordered pair (V, E) where V is a set of elements, called *vertices* and E is a set of unordered pairs of distinct vertices in V, called *edges*. If $(u, v) \in E$, vertices u and v are said to be *adjacent* to each other, and the edge (u, v) is said to be *incident* upon each of u and v. A sequence of vertices $[v_{i_1}, v_{i_2}, \ldots, v_{i_k}]$ is called a *path* from v_{i_1} to v_{i_k} in G if the vertices are all distinct and $(v_{i_j}, v_{i_{j+1}}) \in E$ for $j = 1, 2, \ldots, k-1$. Furthermore, if $v_{i_1} = v_{i_k}$, the path is called a *cycle*. The length of a path (or cycle) is the number of edges on the path (resp., cycle). If the length of a cycle is odd (resp., even), it is called an *odd* (resp., *even*) *cycle*. The degree of vertex v, denoted by $d_G(v)$, is the number of edges incident upon v in G. Let $\Delta(G) = max_{v \in V}\{d_G(v)\}$.

Given a graph $G = (V, E)$, we can partition V into subsets $V_1, V_2, \ldots, V_r$ such that vertices u and v belong to V_i if and only if there is a path from u to v in G. Let $E(V_i) = \{(u, v) \in E \mid u, v \in V_i\}$ for $i = 1, 2, \ldots, r$. The graphs $G_i = (V_i, E(V_i))$ are called *connected components* of G.

A graph $G = (V, E)$ is called a *planar graph* if it can be drawn in the plane in such a way that each vertex in V is represented by a point; each edge $(u, v) \in E$ is represented by a continuous line connecting the two points which represent u and v; and no two lines, which represent edges, share any points, except in their ends. Such a drawing is called a *planar embedding* of G. A graph $G = (V, E)$ is called a *bipartite graph* if V can be partitioned into two nonempty subsets V_1 and V_2 such that $V_1 \cap V_2 = \phi$ and no two vertices in the same subset are adjacent. It is well known [4] that a graph G is bipartite if and only if there is no odd cycle in G.

For a subset $V' \subset V$ of $G = (V, E)$, the graph obtained by deleting from G all vertices in V' and all the edges incident upon them is called a *vertex-deleted subgraph* of G and is denoted by $G^v(V - V')$. Namely, $G^v(V - V') = (V - V', E(V - V'))$ where $E(V - V') = \{(u, v) \in E \mid u, v \in V - V'\}$. Likewise, for a subset $E' \subset E$ of $G = (V, E)$, an *edge-deleted subgraph* of G, denoted by $G^e(E - E')$, is obtained by deleting all edges in E' from G; that is, $G^e(E - E') = (V, E - E')$.

---

[1] The work of this author was supported by Division of Engineering Research, Michigan State University.
[2] The work of these authors was supported in part by National Science Foundation grants MIP-84-51510 and CDR-85-00108, and a grant from AT&T.

G	$\Delta(G)$		
	≤ 2	$= 3$	≥ 4
General	$O(\|V\|)$	NPC	NPC
Planar	$O(\|V\|)$	$O(\|V\|^3)$	NPC

Table 1: Complexity Results for the VDB Problem

Given a graph $G = (V, E)$ and an integer $k \geq 0$, the *vertex-* (resp., *edge-*) *deletion graph bipartization problem*, abbreviated to VDB (resp., EDB), is the problem of finding a set of k or fewer vertices $V' \subset V$ (resp., edges $E' \subset E$) in G such that the subgraph $G^v(V - V')$ (resp., $G^e(E - E')$) is bipartite, or equivalently, free of odd cycles. The *minimum vertex-* (resp., *edge-*) *deletion graph bipatization problem*, abbreviated to MVDB (resp., MEDB), is the problem of finding such a vertex (resp., edge) set of minimum cardinality.

Garey, Johnson and Stockmeyer [7] and Yannakakis [17] proved that the EDB problem is NP-complete even if $G = (V, E)$ is a cubic graph, i.e., $d_G(v) = 3$ for every $v \in V$. On the other hand, Hadlock [8] showed that the MEDB problem is solvable in $O(|V|^3)$ time if the graph is planar. As for the VDB problem, Yannakakis [17] and Krishnamoorthy and Deo [11] showed that it is NP-complete even if G is planar and $\Delta(G) = 6$. Given a graph $G = (V, E)$, the *line graph*, denoted by $G_l = (V_l, E_l)$, of G is defined as follows: There is a one-to-one correspondence between V_l and E. If two edges in E are incident upon the same vertex in G, then there is an edge in E_l which connects the two corresponding vertices in V_l. One can easily show by constructing the line graph of a cubic graph that the VDB problem is NP-complete for a general graph G with $\Delta(G) = 4$.

The first goal of this paper is to present complete complexity results for the VDB problem. We arrive at this by first establishing a relationship between the VDB and EDB problems for a graph $G = (V, E)$ with $\Delta(G) \leq 3$. Namely, we prove that the VDB problem has a solution $V' \subset V$ with $|V'| \leq k$ if and only if the EDB problem has a solution $E' \subset E$ with $|E'| \leq k$. Since the EDB problem is NP-complete for cubic graphs [7,17], this immediately implies that the VDB problem is NP-complete for cubic graphs. Furthermore, since the MEDB problem is solvable in $O(|V|^3)$ time if $G = (V, E)$ is planar [8], this relationship also implies that the MVDB problem is solvable in $O(|V|^3)$ time for the case when G is planar and $\Delta(G) \leq 3$. It should further be noted that if $\Delta(G) \leq 2$, the graph G is always planar and hence the VDB problem is solvable in polynomial time, and in fact, in $O(|V|)$ time. Finally, we prove that the VDB problem becomes NP-complete for a planar graph G if $\Delta(G) = 4$. We give a polynomial transformation from the Planar 3-Satisfiability problem [12] to this problem. The complete complexity results for the VDB problem are summarized in Table 1.

The second goal of this paper is to show the NP-completeness of the *via minimization problem* which arises in the design of integrated circuits and printed circuit boards. Given a set of wire segments and two layers for routing, the problem is to assign the segments to one of the layers so that the number of vias required to electrically connect the segments which are assigned to different layers is minimized.

In 1971, Hashimoto and Stevens [9] first considered this problem for the case of grid based layouts with the maximum "junction" degree being limited to two, where a *grid based layout* is a layout in which all wire segments are placed in parallel to one of the two perpendicular axes, and *"junction" degree* is defined to be the number of wire segments which meet at a single point and which are to be electrically connected. In 1980, Kajitani [10] showed that the problem is solvable in polynomial time for this particular case. Later Chen, Kajitani and Chan [2] and Pinter [16] extended its polynomial solvability for grid-based layouts to the case of the maximum junction degree three. Quite recently, Molitor [13] and Naclerio, Masuda, and Nakajima [15] showed that the via minimization problem is solvable in polynomial time even for general layouts if the maximum junction degree is limited to three.

On the other hand, Naclerio, Masuda, and Nakajima [14] showed that the via minimization decision problem is NP-complete if the maximum junction degree exceeds five. Thus, the cases of the maximum junction degrees 4 and 5 have been left as open problems. In this paper, using the NP-completeness result for the VDB problem for planar graphs, we prove that the via minimization decision problem is NP-complete even when the maximum junction degree is limited to four.

In Section 2, we first show that the MVDB problem for a general graph $G = (V, E)$ is easily solvable in $O(|V|)$ time if $\Delta(G) \leq 2$. We then prove that if $\Delta(G) = 3$, the VDB problem has a solution of size

k or less if and only if the EDB problem has a solution of size k or less. As colloraries of this result, we show that (1) the VDB problem is NP-complete even for cubic graphs, and (2) the MVDB problem is solvable in $O(|V|^3)$ time when G is planar and $\Delta(G) = 3$. Finally, using a transformation from the Planar 3-Satisfiability problem [12], we prove that the VDB problem becomes NP-complete for a planar graph G if $\Delta(G) = 4$. In Section 3, we consider the via minimization problem for two layers. We show how the VDB problem for planar graphs can be transformed to this problem. This leads us to prove that the via minimization decision problem is NP-complete even when the maximum junction degree is limited to four. Section 4 concludes the paper with some further comments on the via minimization problem.

2. GRAPH BIPARTIZATION

In this section, we investigate the computational complexity of the VDB problem from the point of view of vertex degree constraints. We first show that there is a close relationship between the VDB and EDB problems for a graph $G = (V, E)$ with $\Delta(G) \leq 3$. If $\Delta(G) \leq 2$, each connected component of the graph G is either a single vertex, a path or a cycle. Therefore, the MVDB and MEDB problems can be solved by first finding all odd cycles and then selecting an arbitrary vertex and edge, respectively, from each such cycle. This can easily be done in $O(|E|)$ time, which is in fact, $O(|V|)$ time, since G is planar in this case. We now consider the case of $\Delta(G) = 3$, and establish a close relationship between the VDB and EDB problems.

Theorem 1. *For any graph $G = (V, E)$ with $\Delta(G) = 3$, there exists a subset $E' \subset E$ such that $|E'| \leq k$ and $G^e(E - E')$ is bipartite if and only if there exists a subset $V' \subset V$ such that $|V'| \leq k$ and $G^v(V - V')$ is bipartite.*

Proof. Suppose that there exists a subset $E' \subset E$ with $|E'| \leq k$ such that $G^e(E - E')$ is bipartite. Let $V' = \{v \in V \mid \textit{for each edge } (u, w) \in E', v \textit{ is either } u \textit{ or } w\}$. It is clear that $|V'| \leq |E'| \leq k$ and $G^v(V - V')$ is bipartite.

Conversely, suppose that there exists a subset $V' \subset V$ with $|V'| \leq k$ such that $G^v(V - V')$ is bipartite. First, we construct, if necessary, a new graph $\hat{G} = (\hat{V}, \hat{E})$ from $G = (V, E)$ such that (1) for any pair of vertices $u, w \in V'$, $(u, w) \notin \hat{E}$ and (2) $\hat{G}^v(\hat{V} - V')$ is bipartite. Let $E_p = \{(u, w) \in E \mid u, w \in V'\}$. If $E_p = \phi$, we do nothing, *i.e.*, $\hat{G} = G$. Otherwise, we replace each edge $(u, w) \in E_p$ by a path of length three $[u, u', w', w]$ in $\hat{G}$. More precisely, $\hat{V} = V \cup \{u', w' \mid (u, w) \in E_p\}$ and $\hat{E} = E - E_p \cup \{(u, u'), (u', w'), (w', w) \mid (u, w) \in E_p\}$. Note that any odd (resp., even) cycle containing $(u, w) \in E_p$ in G remains odd (resp., even) in $\hat{G}$, and hence $\hat{G}^v(\hat{V} - V')$ remians bipartite. Furthermore, $\Delta(\hat{G}) = 3$.

Next, we obtain, from V', a subset $\hat{E}' \subset \hat{E}$ such that $|\hat{E}'| \leq k$ and $\hat{G}^e(\hat{E} - \hat{E}')$ is bipartite. Since $\hat{G}^v(\hat{V} - V')$ is bipartite and for any two vertices $u, w \in V'$, $(u, w) \notin \hat{E}$, the vertex set $\hat{V}$ can be partitioned into three mutually disjoint subsets, V', X, and Y such that no two vertices in the same subset are adjacent.

We first remove from V' those vertices v with $d_{\hat{G}}(v) = 1$ because the edges incident upon v are not in any cycle in $\hat{G}$. Let V'' denote the set of the remaining vertices. It is not difficult to see, as illustrated in Fig. 1 (a), that an odd cycle is formed only if there exists a vertex in V'' which is adjacent to a vertex in X and a vertex in Y simultaneously. Otherwise, as can be seen from Fig. 1 (b), all cycles are of even length since $\hat{G}^v(V - V') = \hat{G}^v(X \cup Y)$ is bipartite. Note that if not all of the vertices adjacent to a vertex v in V'' are in one of the two sets X and Y, then either X or Y contains exactly one vertex, say w, since $d_{\hat{G}}(v) \leq 3$. It is clear that if we delete this edge (v, w), all odd cycles passing through v will be removed. Therefore, we select all such edges (v, w) as members of $\hat{E}'$. Then $\hat{G}^e(\hat{E} - \hat{E}')$ is bipartite and $|\hat{E}'| \leq |V''| \leq |V'| \leq k$.

Finally, from $\hat{E}'$ we construct a subset $E' \subset E$ such that $|E'| \leq k$ and $G^e(E - E')$ is bipartite. Let $e \in \hat{E}'$. If $e \in E$, we put e into E'. If $e \notin E$, then e must be of the form (u, u') such that $u \in V' \subset V$ and $u' \in \hat{V} - V$. In this case, there is a vertex $w \in V' \subset V$ such that $(u, w) \in E$, and we put this edge (u, w) into E'. Since the deletion of the edge (u, u') from $\hat{G}$ has the same effect on the elimination of cycles as that of its corresponding unique path of length three from u to w in $\hat{G}$, it is clear that the deletion of the edge (u, w) from G eliminates all cycles passing through u. Therefore, $G^e(E - E')$ is bipartite, and furthermore, $|E'| \leq |\hat{E}'| \leq k$. This completes the proof of the theorem. □

Garey, *et. al* [7] and Yannakakis [17] showed that the EDB problem is NP-complete even if a graph $G = (V, E)$ is a cubic graph, *i.e.*, $d(v) = 3$ for every $v \in V$. Thus, we can derive the following result from Theorem 1.

Corollary 1. *The VDB problem is NP-complete even for a cubic graph.* □

On the other hand, if the graph $G = (V, E)$ is planar, Hadlock [8] showed that the MEDB problem can be solved in $O(|V|^3)$ time. Therefore, we obtain the following result from Theorem 1.

Corollary 2. *For a planar graph $G = (V, E)$ with $\Delta(G) = 3$, the MVDB problem is solvable in $O(|V|^3)$ time.* □

Remark 1. The above discussion suggests a way to solve the MVDB problem for a planar graph G with $\Delta(G) = 3$. That is, we first solve the MEDB problem using Hadlock's approach [8], and then convert its solution to a solution to the MVDB problem. We present a more direct approach to the MVDB problem. This approach can also be used to obtain an approximate solution for the case of $\Delta(G) \geq 4$.

Let $G = (V, E)$ be a given planar graph. Without loss of generality, we can assume that G is *biconnected*, that is, for every pair of vertices u and v in V, there are at least two vertex disjoint paths from u to v.

Let $\tilde{G}$ be a planar embedding of G. Since G is biconnected, $\tilde{G}$ partitions the rest of the plane into a number of connected regions. The closures of those regions are called the *faces* of $\tilde{G}$. Let F be a set of such faces. F includes a special face called an *exterior face* which represents the infinite region outside the embedding $\tilde{G}$. Note that each face in F corresponds to a cycle in G, which is called a *fundamental cycle*. We create a new graph $G^n = (V^n, E^n)$, where $V^n = V \cup F$ and $E^n = \{(v, f) \mid v \in V$ *is on the fundamental cycle corresponding to* $f \in F\}$. Then we follow Hadlock [8] and find a pairing of odd degree vertices (or faces) in F such that the total sum of lengths of shortest paths between such pairs is a minimum. □

We now consider the case in which $G = (V, E)$ is planar and $\Delta(G) = 4$. We prove that the VDB problem becomes NP-complete for this case.

Theorem 2. *The VDB problem is NP-complete for a planar graph even if $\Delta(G) = 4$.*

Proof. Since the VDB problem belongs to the class NP, it suffices to show that a known NP-complete problem is transformable in polynomial time to this problem. We start with the following problem which was shown to be NP-complete by Licktenstein [12].

Planar 3-Satisfiability (P3SAT)

Instance: A set $U = \{v_i \mid 1 \leq i \leq n\}$ of n Boolean variables and a set $C = \{c_j \mid 1 \leq j \leq m\}$ of m clauses over U such that each clause c_j contains exactly three variables or their complements. Furthermore, the following graph is planar:

$G_C = (V_C, E_C)$, where

$V_C = \{c_j \mid 1 \leq j \leq m\} \cup \{v_i \mid 1 \leq i \leq n\}$ and

$E_C = \{(c_j, v_i) \mid v_i \in c_j \text{ or } \bar{v}_i \in c_j\} \cup \{(v_i, v_{i+1}) \mid 1 \leq i < n\} \cup \{(v_n, v_1)\}$.

Question: Is C satisfiable? Namely, is there a truth assignment for U such that each clause in C is true?

Given $U = \{v_i \mid 1 \leq i \leq n\}$ and $C = \{c_j \mid 1 \leq j \leq m\}$ together with a planar embedding of $G_C = (V_C, E_C)$, we construct a planar graph $G = (V, E)$ in the following way. The graph contains two kinds of components: clause components and variable components. And each of them is placed in the region in which the corresponding vertex $c_j \in V_C$ or $v_i \in V_C$ is drawn. For each clause $c_j \in C$ we create a clause component of four vertices $w_j^1, w_j^2, w_j^3, w_j^4$ and one edge (w_j^1, w_j^4) as shown in Fig. 2 (a). As will be explained later, each pair of vertices w_j^l and w_j^{l+1} for $l = 1, 2, 3$, will be connected to a pair of vertices of an appropriate variable component. For each variable $v_i \in U$ we construct a variable component of $8n_i$ vertices and $12n_i$ edges, where n_i is the number of times variable v_i or its complement $\bar{v}_i$ appears in C. As illustrated in Fig. 2(b), each variable component is made up of $4n_i$ triangles whose bottom edges form a cycle of length $4n_i$. The vertices on this cycle correspond to v_i and $\bar{v}_i$, alternately. Namely, b_i^{k1} and b_i^{k3} correspond to v_i and b_i^{k2} and b_i^{k4} correspond to $\bar{v}_i$ for $k = 1, 2, \ldots, n_i$. Note that a group of four consecutive triangles, more precisely, eight vertices a_i^{kl} and b_i^{kl} for $l = 1, 2, 3, 4$ correspond to variable v_i. If $\bar{v}_i \in c_j$ (resp., $v_i \in c_j$), then the two top vertices a_i^{k1} and a_i^{k2} (resp., a_i^{k2} and a_i^{k3}) for some $k \in \{1, 2, \ldots, n_i\}$ are connected to a pair of vertices w_j^l and w_j^{l+1} for some $l \in \{1, 2, 3\}$ of the clause component corresponding to c_j. For example, see Fig. 3. It is easy to see that $|V| = 4m + \sum_{i=1}^{n} 8n_i = 28m$ and $|E| = 7m + \sum_{i=1}^{n} 12n_i = 43m$. Therefore, this transformation is done in polynomial time. Furthermore, it is clear that G is planar and $\Delta(G) = 4$.

We now prove that there is a truth assignment for U such that each clause $c_j \in C$ is true if and only if there is a subset $V' \subset V$ of $G = (V, E)$ such that $|V'| = 6m$ and $G^v(V - V')$ is bipartite.

Suppose that there is a subset $V' \subset V$ such that $|V'| = 6m$ and $G^v(V - V')$ is bipartite. Since each variable component contains $4n_i$ triangles, which are odd cycles, we must delete at least $2n_i$ vertices to break these triangles. This can be done only if every other vertex on the cycle of length $4n_i$ is so chosen. More precisely, we must delete n_i pairs of vertices either b_i^{k1} and b_i^{k3} or b_i^{k2} and b_i^{k4} for $k = 1, 2, \ldots, n_i$, for each $i = 1, 2, \ldots, n$. Since $\sum_{i=1}^{n} 2n_i = 6m$, no other vertices are deleted. Therefore, we can make a consistent assignment of value *true* or *false* to each variable v_i in the following manner: If vertex b_i^{12} (resp., b_i^{13}) is deleted from the variable component corresponding to v_i, assign *false* (resp., *true*) to v_i, for $i = 1, 2, \ldots, n$.

Note that all four vertices in each clause component belong to a cycle of length 13 which connects those four vertices and three vertices, one from each of the three corresponding variable components. We call such a cycle a *clause cycle*. For example, a cycle $[w_2^1, a_4^{1\,3}, b_4^{1\,3}, a_4^{1\,2}, w_2^2, a_2^{2\,2}, b_2^{2\,2}, a_2^{2\,1}, w_2^3, a_3^{2\,3}, b_3^{2\,3}, a_3^{2\,2}, w_4^2, w_2^1]$ is a clause cycle for clause c_2 in Fig. 3. Since by assumption, each such cycle is broken, at least one vertex labeled b in the three variable components must be deleted. Since the variable or its complement corresponding to such a vertex is assigned *true*, the clause corresponding to this cycle of length 13 is *true*. Therefore, there is a truth assignment for U such that each clause in C is true.

On the other hand, suppose that there is a truth assignment for U such that each clause c_j in C is true. If v_i is assigned *true* (resp., *false*), then delete n_i pairs of vertices b_i^{k2} and b_i^{k4} (resp., b_i^{k1} and b_i^{k3}) for $k = 1, 2, \ldots, n_i$, for each $i = 1, 2, \ldots, n$. As mentioned before, the removal of these vertices breaks all $4n_i$ triangles and leaves $2n_i$ paths of length 2 in the variable component corresponding to v_i. It also breaks all clause cycles. Furthermore, it is clear that all the other (possibly odd) cycles are eliminated because each variable component is chopped into $2n_i$ paths of length 2. Therefore, the remaining graph dose not contain any odd cycles and hence it is bipartite. This completes the proof. □

3. VIA MINIMIZATION

In this section, using Theorem 2, we prove that the via minimization decision problem for *two* layers is NP-complete even if the maximum junction degree is limited to four. We start with some definitions.

A *circuit* is specified by a set of modules M, a set of terminals T and a set of nets N. The *terminals* in T are located on the boundary of the *modules* in M and each *net* specifies which terminals are to be electrically connected. Such connections are made by patterning conductive paths on one of two *layers*. Such paths are made up of straight line segments, called *wire segments*. We assume that the terminals are available on both layers and each wire segment can be assigned to either layer. Note that the vertical projection of each wire segment is fixed in the plane but its layer assignment is not specified. A point other than a terminal at which two or more wire segments meet and are electrically connected is called a *junction*. A wire segment is said to be *incident* upon a junction at which it meets. Wire segments which are incident upon the same junction are said to be *adjacent* to each other, and the number of such segments is called the *junction degree*. If wire segments incident upon the same junction are assigned to different layers, a *via* is placed at the junction to electrically connect them. If the vertical projections of two wire segments that are not electrically connected intersect, they are said to *cross* each other. A layer assignment is said to be *valid* if no two wire segments that cross each other are assigned to the same layer and no two adjacent wire segments are assigned to different layers without a via.

The *via minimization decision problem* (abbreviated to VM) for two layers is defined as follows:

Via Minimization (VM)

Instance: A set $M = \{m_i \mid 1 \leq i \leq p\}$ of modules, a set $T = \{t_i \mid 1 \leq i \leq q\}$ of terminals, a set $W = \{w_i \mid 1 \leq i \leq r\}$ of wire segments whose vertical projections are fixed in the plane, and an integer $k \geq 0$.

Question: Is there a valid layer assignment for W which requires k or fewer vias using two layers?

In order to show a transformation from the VDB problem for planar graphs to the VM problem, we will use the sublayout L shown in Fig. 4. L has a single module m, two terminals t and t' on its boundary, and wire segments w and w' which connect terminals t and t' to junctions j and j', respectively. Note that the terminals t and t' belong to different nets and thus the wires w and w' must be assigned to different layers.

Let $G = (V, E)$ be a planar graph. A *straight line planar embedding* of G is a planer embedding of G in which every edge in G is represented by a straight line segment. It is known [3,5] that such an embedding of G can be obtained in polynomial time, and we will denote it by $\bar{G}$. For simplicity, we call a straight line segment in $\bar{G}$ which represents an edge (x, y) in G, a *segment* (x, y), and the endpoints of the line segment which represent vertices x and y, *points* x and y, respectively.

We first create a small region surrounding each segment (x, y) in $\bar{G}$ so that no two such regions overlap. We then replace each segment (x, y) by a sublayout L in such a manner that L is completely within the region surrounding (x, y) and junctions j and j' coincide with points x and y, respectively.

We denote the resultant layout by $L(G)$. Fig. 5 shows an example graph G_1 and its corresponding layout $L(G_1)$. Note that for each cycle in G, there is a cycle of sublayouts or modules in $L(G)$. We call such a cycle of modules an *m-cycle*. If the number of modules in an m-cycle is odd (resp., even), it is called an *odd* (resp., *even*) m-cycle. Consider the m-cycle consisting of modules m_1, m_2 and m_3 shown in Fig. 5(b). Suppose that wire segment w_1 is assigned to layer 1. Then w_1' must be assigned to layer 2. To avoid a via at junction j_2, w_2 should be assigned to layer 2. Consequently, w_2' must be assigned to layer 1 and hence w_3' should be assigned to layer 1. This implies that w_3 must be assigned to layer 2. In order to electrically connect the wire segments w_3 and w_1, a via needs to be placed at junction j_1. It is not difficult to see that a via is always required to have a valid layer assignment if there is an odd m-cycle in $L(G)$.

On the other hand, suppose that there is no odd m-cycle in $L(G)$. Then, a valid layer assignment for $L(G)$ which requires no via will be obtained in the following manner. Assign an arbitrary wire segment and all of its adjacent wire segments to layer 1. Find wire segments which cross the wire segments just assigned to layer 1. Assign those segments to layer 2. Then find unassigned wire segments which cross the segments just assigned to layer 2 and assign them to layer 1. Repeat this process until all segments are assigned to one of the layers. Since there is no odd m-cycle, no conflict on layer assignment would occur. Therefore, we have the following lemma.

Lemma 1. *There exists a valid layer assignment for layout $L(G)$ which requires no via if and only if it is free of odd m-cycles.* □

We are now ready to show the NP-completeness of the VM problem even when the maximum junction degree is limited to four.

Theorem 4. *The VM problem is NP-complete even when the maximum junction degree is limited to four.*

Proof. It is easy to see that the VM problem belongs to the class NP. Thus, it suffices to show that the VDB problem for a planar graph G with $\Delta(G) = 4$ is transformable in polynomial time to the VM problem.

Let $G = (V, E)$ be a planer graph such that $\Delta(G) = 4$ and k be a nonnegative integer. We construct a layout $L(G)$ in the manner described above.

Suppose that there is a set of vertices $V' \subset V$ such that $|V'| \leq k$ and $G^v(V - V')$ is bipartite. Let J be a set of junctions in $L(G)$ which correspond to the vertices in V'. We first delete from $L(G)$ all junctions in J and the wire segments incident upon them. Let $L^d(G)$ denote the resultant layout. Fig. 6 (a) depicts such a layout which is obtained by deleting junction j_1 from $L(G_1)$ of Fig. 5 (b). Note that there are two types of "*degenerated*" sublayouts in $L^d(G)$. Sublayouts of Type 1 (resp., 0) are those that consist of a module and one (resp., no) wire segment. In Fig. 6 (a), sublayouts containing modules m_1, m_3, m_6, and m_8 are of Type 1. If junction j_2 were also deleted, the sublayout containing module m_1 would be of Type 0. We then delete those degenerated sublayouts from $L^d(G)$. Let $L'(G)$ be the resultant layout. Since $L'(G)$ does not contain any odd m-cycle, by Lemma 1 there exists a valid layer assignment for $L'(G)$ which requires no via. Fig. 6 (b) illustrates such a layer assignment for $L'(G_1)$. Then, based on this layer assignment, we can obtain a valid layer assignment for $L^d(G)$ as follows: For each degenerated sublayout of Type 1, assign its only wire segment to the same layer as all of its adjacent wire segments in $L'(G)$. Such a layer assignment for $L'(G_1)$ is shown in Fig. 6 (c).

We now find layer assignments for those wire segments which are incident upon the junctions in J. For each wire segment which is missing from some degenerated sublayout of Type 1, if the remaining wire segment is assigned to layer 1, we assign the missing segment to layer 2 and vice versa. We assign a pair of wire segments in each degenerated sublayout of Type 0 to different layers arbitrarily. Because vias are placed at those junctions in J, necessary electrical connections will be made. Therefore, a valid layer assignment exists for $L(G)$ which requires k or fewer vias, since $|J| \leq k$. A final valid layer assignment for $L(G_1)$ is shown in Fig. 6 (d).

Conversely, suppose that there is a set of k or fewer vias for which a valid layer assignment for $L(G)$ exists. Let J be the set of junctions at which the vias are placed. Let V' be the set of vertices which correspond to the junctions in J.

Consider the vertex-deleted subgraph $G^v(V - V')$ and its corresponding layout $L(G^v(V - V'))$. It is not difficult to see that $L(G^v(V - V'))$ is obtained by deleting all such sublayouts that contain at least one wire segment which is incident upon some junction in J. Since the layer assignment for $L(G^v(V - V'))$ is valid and requires no via, by Lemma 1 there is no odd m-cycle in $L(G^v(V - V'))$. This implies that there is no odd cycle in $G^v(V - V')$ and hence $G^v(V - V')$ is bipartite. Therefore, there exists a set of vertices V' such that $|V'| = |J| \leq k$ and $G^v(V - V')$ is bipartite.

We have proved that there is a set of vertices $V' \subset V$ such that $|V'| \leq k$ and $G^v(V - V')$ is bipartite if and only if a valid layer assignment exists for $L(G)$ which requires k or fewer vias. Since the construction of $L(G)$ only requires replacing each segment in $\bar{G}$ with a sublayout L, it can easily be accomplished in polynomial time. Furthermore, the number of wire segments incident upon a junction in $L(G)$ is the same as the number of segments incident upon the corresponding point in $\bar{G}$. Since $\Delta(G) = 4$, the maximum junction degree in $L(G)$ is limited to four. This completes the proof of the theorem. □

Remark 2. Using the same arguments as in Naclario, Masuda, and Nakajima [14], one can establish the NP-completeness of the VM problem for two layers under any combination of the following two constraints as long as the maximum junction degree is limited to four or more:

1. The layout is restricted to be grid based.
2. Vias are placed only at the junctions which existed in the input layout.

4. CONCLUSION

We have presented complexity results for the vertex-deletion graph bipartization (VDB) problem. These results completely close the gap between the polynomially solvable cases and NP-complete cases for general and planar graphs as shown in Table 1. We have also shown that the via minimization decision problem is NP-complete for two layers when the maximum junction degree is limited to four. Quite recently, Molitor [13] showed that the problem of assigning wire segments to three or more layers without using vias is NP-complete when the maximum junction degree is limited to four. Since the via minimization problem for two layers is solvable in polynomial time when the maximum junction degree is limited to three [2,13,15,16], our result has completely settled the complexity issues on via minimization.

References

[1] K. Aoshima and M. Iri, "Comments on F. Hadlock's Paper: Finding a Maximum Cut of a Planar Graph in Polynomial Time," *SIAM Journal on Computing*, Vol. 6, No. 1, pp. 86–87, March 1977.

[2] R-W. Chen, Y. Kajitani and S-P. Chan, "A Graph-Theoretic Via Minimization Algorithm for Two-Layer Printed Circuit Boards," *IEEE Trans. on Circuits and Systems*, Vol. CAS-30, No. 5, pp. 284–299, May 1983.

[3] N. Chiba, T. Nishizeki, S. Abe and T. Ozawa, "A Linear Time Algorithm for Embedding Planar Graphs using PQ-trees," *Journal of Computer and System Sciences*, Vol. 30, pp. 54–76, 1985.

[4] S. Even, *Graph Algorithms*, Computer Science Press, Rockville, MD, 1979.

[5] I. Fary, "On Straight Line Representation of Planar Graphs," *Acta Sci. Math. (Szeged)*, Vol. 11, pp. 229–233, 1948.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, San Francisco, CA, 1979.

[7] M. R. Garey, D. S. Johnson and L. Stockmeyer, "Some Simplified NP-complete Graph Problems," *Theoretical Computer Science*, Vol. 1, pp. 237–267, 1976.

[8] F. Hadlock, "Finding a Maximum Cut of a Planar Graph in Polynomial Time," *SIAM Journal on Computing*, Vol. 4, No. 3, pp. 221–225, Sept. 1975.

[9] A. Hashimoto and J. Stevens, "Wire Routing by Optimizing Channel Assignment within Large Apertures," *Proc. 8th Design Automation Workshop*, Atlantic City, NJ, June 1971, pp. 155–169.

[10] Y. Kajitani, "On Via Hole Minimization of Routing on a 2-Layer Board," *Proc. 1980 IEEE International Conf. on Circuits and Computers*, New York, NY, Oct. 1980, pp. 295–298.

[11] M. S. Krishnamoorthy and N. Deo, "Node-Deletion NP-Complete Problems," *SIAM Journal on Computing*, Vol. 8, No. 4, pp. 619–625, Nov. 1979.

[12] D. Lichtenstein, "Planar Formulae and their Uses," *SIAM Journal on Computing*, Vol. 11, No. 2, pp. 329–343, May 1982.

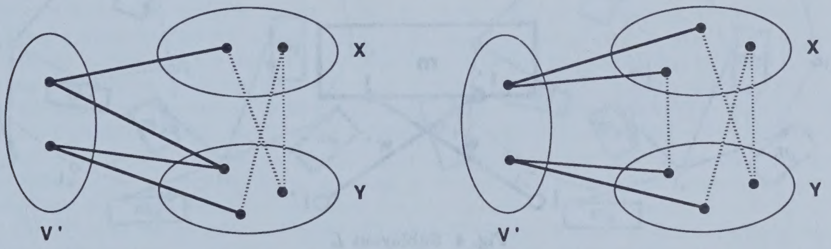
[13] P. Molitor, "On the Contact-minimization Problem," *Proc. 4th Annual Symp. on Theoretical Aspects of Computer Science*, Passau, West Germany, Feb. 1987; published as *Lecture Notes in Computer Science*, No. 247, Springer-Verlag, pp. 420–431.

[14] N. J. Naclerio, S. Masuda and K. Nakajima, "Some NP-completeness Results for Via Minimization," *Proc. 1987 Conf. on Information Sciences and Systems*, Baltimore, MD, March 1987, pp. 611–616.

[15] N. J. Naclerio, S. Masuda and K. Nakajima, "Via Minimization for Gridless Layouts," *Proc. 24th ACM/IEEE Design Automation Conference*, Miami Beach, FL, June 1987, pp. 159–165.

[16] R. P. Pinter, "Optimal Layer Assignment for Interconnect," *Proc. International Conf. on Circuits and Computers*, New York, NY, Sept. 1982, pp. 398–401.

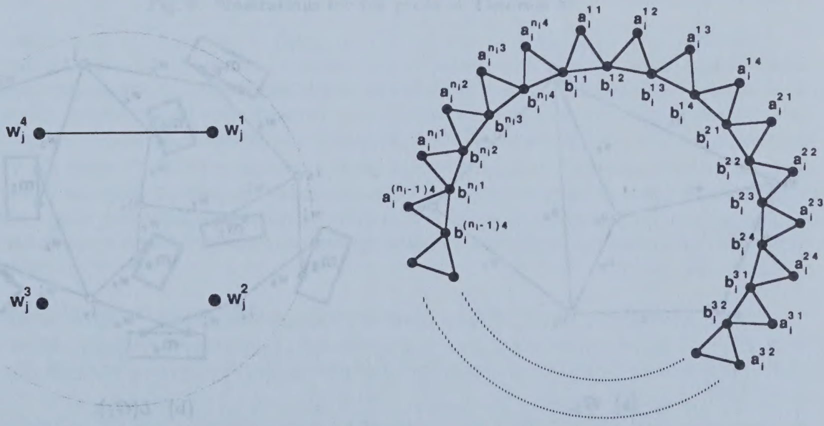
[17] M. Yannakakis, "Node- and Edge-deletion NP-complete Problems," *Proc. 10th Annual ACM Symp. on Theory of Computing*, San Diego, CA, Oct. 1978, pp. 253–264.



(a) A case in which an odd cycle is formed.

(b) A case in which an even cycle is formed.

Fig. 1. Formation of cycles.


w_j^4
w_j^1
w_j^3
w_j^2
a_i^{11}
a_i^{12}
a_i^{13}
a_i^{14}
$a_i^{n_i4}$
$a_i^{n_i3}$
$a_i^{n_i2}$
$a_i^{n_i1}$
$a_i^{(n_i-1)4}$
b_i^{11}
b_i^{12}
b_i^{13}
b_i^{14}
$b_i^{n_i4}$
$b_i^{n_i3}$
$b_i^{n_i2}$
$b_i^{n_i1}$
$b_i^{(n_i-1)4}$
a_i^{21}
a_i^{22}
a_i^{23}
a_i^{24}
b_i^{21}
b_i^{22}
b_i^{23}
b_i^{24}
b_i^{31}
b_i^{32}
a_i^{31}
a_i^{32}

(a) Clause component.

(b) Variable component.

Fig. 2. Clause and variable components.

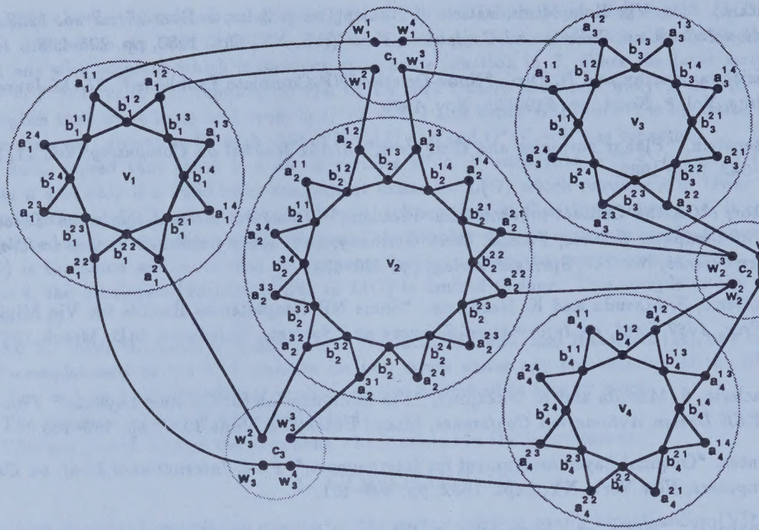

w_1^1
w_1^4
c_1
w_1^3
w_1^2
a_1^{11}
a_1^{12}
b_1^{12}
b_1^{11}
b_1^{13}
a_1^{13}
a_1^{24}
b_1^{24}
v_1
b_1^{14}
a_1^{23}
a_1^{14}
b_1^{23}
b_1^{22}
b_1^{21}
a_1^{22}
a_1^{21}
a_2^{12}
a_2^{13}
a_2^{11}
b_2^{12}
b_2^{13}
b_2^{14}
a_2^{14}
b_2^{11}
b_2^{21}
a_2^{21}
a_2^{34}
b_2^{34}
v_2
b_2^{22}
a_2^{22}
a_2^{33}
b_2^{33}
b_2^{23}
a_2^{32}
b_2^{32}
b_2^{31}
b_2^{24}
a_2^{23}
a_2^{31}
a_2^{24}
a_3^{12}
a_3^{13}
b_3^{13}
a_3^{11}
b_3^{12}
b_3^{14}
a_3^{14}
v_3
b_3^{11}
b_3^{21}
a_3^{24}
a_3^{21}
b_3^{24}
b_3^{22}
b_3^{23}
a_3^{22}
a_3^{23}
w_2^4
w_2^3
c_2
w_2^2
w_2^1
a_4^{11}
a_4^{12}
b_4^{12}
b_4^{11}
b_4^{13}
a_4^{24}
a_4^{13}
b_4^{24}
v_4
b_4^{14}
a_4^{23}
a_4^{14}
b_4^{23}
b_4^{21}
b_4^{22}
a_4^{21}
a_4^{22}
w_3^3
w_3^2
c_3
w_3^1
w_3^4

Fig. 3. Example. $C = \{c_1, c_2, c_3\}$; $c_1 = \{\bar{v}_1, v_2, \bar{v}_3\}$, $c_2 = \{\bar{v}_2, v_3, v_4\}$, $c_3 = \{\bar{v}_1, \bar{v}_2, v_4\}$.

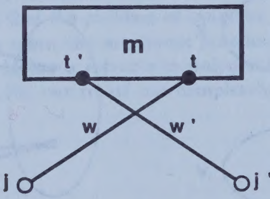
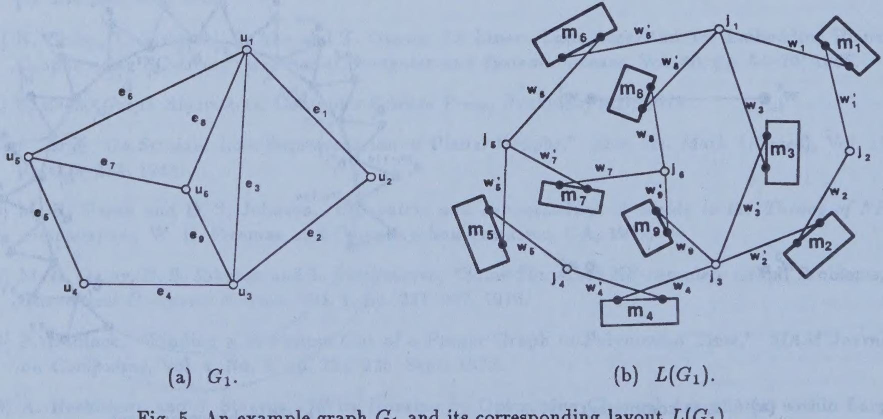


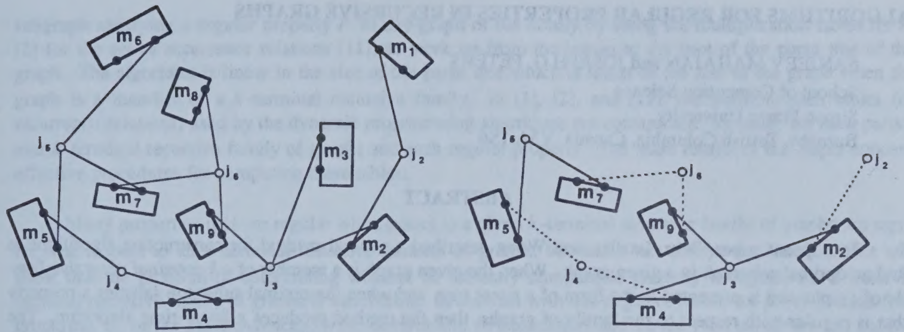
Fig. 4. Sublayout L


u_1
e_6
e_8
e_1
u_5
e_7
u_6
e_3
u_2
e_5
e_9
e_2
u_4
e_4
u_3
m_6
w_6'
j_1
w_1
m_1
w_8'
w_6
m_8
w_3
w_1'
w_8
j_5
m_3
j_2
w_7'
w_7
j_6
w_5'
m_7
w_9'
w_2
w_3'
m_5
m_9
w_5
w_9
m_2
w_2'
j_4
j_3
w_4'
w_4
m_4

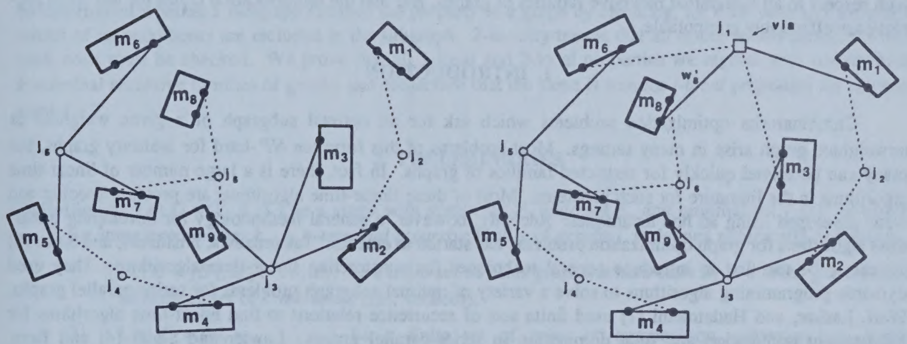
(a) G_1.

(b) $L(G_1)$.

Fig. 5. An example graph G_1 and its corresponding layout $L(G_1)$.


m6
m1
m8
m3
j5
j2
j6
m7
m5
m9
m2
j4
j3
m4
j5
j6
j2
m7
m5
m9
m2
j4
j3
m4

(a) $L(G_1^d)$. (b) A valid layer assignment for $L(G_1')$.


m6
m1
m8
j5
m3
j2
j6
m7
m5
m9
m2
j4
j3
m4
j1
via
w'8
m6
m1
m8
j5
m3
j2
j6
m7
m5
m9
m2
j4
j3
m4

(c) A valid layer assignment for $L(G_1^d)$. (d) A valid layer assignment for $L(G_1)$.

Fig. 6. Illustrations for the proof of Theorem 3.

ALGORITHMS FOR REGULAR PROPERTIES IN RECURSIVE GRAPHS

SANJEEV MAHAJAN and JOSEPH G. PETERS
School of Computing Science
Simon Fraser University
Burnaby, British Columbia, Canada V5A 1S6

ABSTRACT

In a recent paper, Bern, Lawler, and Wong described a general method for constructing algorithms to find an optimal subgraph in a given graph. When the given graph is a member of a *k-terminal recursive* family of graphs and is presented in the form of a *parse tree*, and when the optimal subgraph satisfies a property that is *regular* with respect to the family of graphs, then the method produces a linear-time algorithm. The algorithms assume the existence of *multiplication tables* which are specific to the regular property and to the family of graphs. In this paper we investigate the computation of these multiplication tables. We first show that the general problem of computing these tables is unsolvable. We then show that the properties *vertex independence, vertex domination, irredundance,* and *vertex j-independence* are regular with respect to all k-terminal recursive families of graphs and give effective procedures for constructing multiplication tables for these properties. Finally, we show that these four properties are *local,* that every local property is regular with respect to all k-terminal recursive families of graphs, and that the multiplication tables for any local property are effectively computable.

1. INTRODUCTION

Combinatorial optimization problems which ask for an optimal subgraph of a given weighted or unweighted graph arise in many settings. Most problems of this form are NP-hard for arbitrary graphs, but many can be solved quickly for restricted families of graphs. In fact, there is a large number of linear-time algorithms in the literature for such problems. Most of these linear-time algorithms are problem-specific and were developed using ad hoc techniques. Recently, however, a general methodology for developing linear-time algorithms for graph optimization problems has started to emerge. Takamizawa, Nishizeki, and Saito [8] appear to be the first to introduce general techniques for constructing linear-time algorithms. They used dynamic programming algorithms to solve a variety of optimal subgraph problems for series-parallel graphs. Pfaff, Laskar, and Hedetniemi [7] used finite sets of recurrence relations to find linear-time algorithms for independent domination and total domination in series-parallel graphs. Lawler and Slater [4] and Bern, Lawler, and Wong [1, 2] extended the theory by observing that most of these linear-time algorithms had been found for families of graphs which are *decomposable* or *k-terminal recursive.* Very recent results are reported in [9], [10], and [12]. Wimer, Hedetniemi, and Laskar [11] give a good introduction and a more detailed history of the development of the theory of linear-time graph algorithms.

Intuitively, a family of graphs is *recursive* if every graph of the family can be generated by a finite number of applications of *composition operations* starting with a finite set of *basis* graphs. If each graph in a recursive family of graphs has a set of k distinguished vertices called *terminals,* and each composition operation is defined in terms of certain primitive operations on terminals, then the family of graphs is *k-terminal recursive.* Let Γ be a k-terminal recursive family of graphs. Bern, Lawler, and Wong [2] define *regular properties* in terms of finite range homomorphisms on graph × subgraph pairs (G,H) where $G \in \Gamma$ and H is a subgraph of G. The idea is to extend the composition operations from graphs to equivalence classes of graph × subgraph pairs. A property P is *regular* with respect to a class of graphs if each extended composition operator can be represented as a finite binary *multiplication table.* (Equivalently, a finite set of recurrence relations can be used [11].)

Any member of a k-terminal recursive family of graphs can be represented as a *parse tree* in which the leaves are basis graphs and the internal nodes represent applications of composition operations to the subgraphs represented by their children. A dynamic programming algorithm can now be used to find an optimal

---

This research was supported by the Natural Sciences and Engineering Research Council of Canada under Grant No. A-0322.

subgraph satisfying a regular property P in any graph of the family by using the multiplication tables for P [2] (or the set of recurrence relations [11]) to work up from the leaves to the root of the parse tree of the graph. The algorithm is linear in the size of the parse tree which is linear in the size of the graph when the graph is a member of a k-terminal recursive family. In [1], [2], and [11], the multiplication tables (or recurrence relations) used by the dynamic programming algorithms are constructed "by hand" for each particular k-terminal recursive family of graphs and each regular property. The main results of this paper concern effective procedures for computing these tables.

Many properties that are regular with respect to a given k-terminal recursive family of graphs are regular with respect to all k-terminal recursive families of graphs, but this is not always the case [5]. We will show that the problem of determining whether an arbitrary computable property is regular for a fixed k-terminal recursive family of graphs is unsolvable. In contrast with this negative result, we prove that the four properties of *vertex independence, vertex domination, irredundance,* and *vertex j-independence* are regular with respect to all k-terminal recursive families of graphs. (Two vertices in a graph are *j-independent* if the shortest path between them has length at least j.) We also give effective procedures for constructing multiplication tables for the properties.

This paper concludes with an attempt to characterize properties, such as the four mentioned above, that are regular with respect to all k-terminal recursive families of graphs. Roughly, a property is *1-local* if it can be determined whether a subgraph satisfies the property in a graph by checking, for each vertex of the graph, which of its neighbours are included in the subgraph. *2-locality* means that all nodes within distance 2 from each node must be checked. We prove that all 1-local and 2-local properties are regular with respect to all k-terminal recursive families of graphs and conjecture that the same is true for *j-local* properties for any constant $j > 2$.

2. PRELIMINARIES

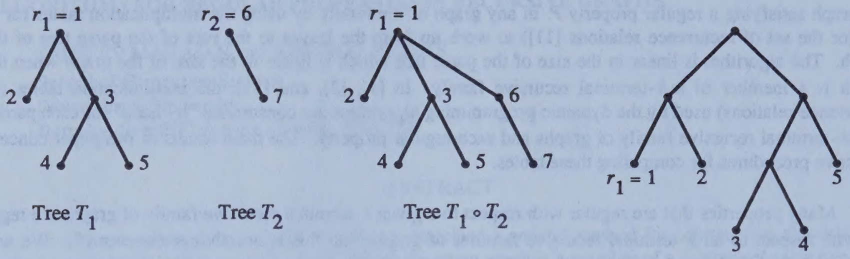
Definition 1. Let $G=(V,E,T)$ be a graph with vertex set V, edge set E, and terminal set $T \subseteq V$ of size k, for some fixed integer k. A *k-terminal recursive* family of graphs Γ is defined recursively as follows.

1. Let $B=\{B_1, B_2, \ldots, B_l\}$ be a finite set of basis graphs, where each B_i is a finite graph having an ordered set of k *terminals* (i.e., vertices).
2. Let $O=\{\circ_1, \circ_2, \ldots, \circ_m\}$ be a finite set of rules of composition, whereby two graphs $G_i=(V_i,E_i,T_i)$ and $G_j=(V_j,E_j,T_j)$ can be combined to produce new graphs $H=G_i \circ_r G_j$; $1 \le r \le m$. Each rule of composition must be defined in terms of the following primitive operations:
   (i) identify a terminal in T_i with a terminal in T_j
   (ii) add an edge between a terminal in T_i and a terminal in T_j
   (iii) select an ordered set of k vertices from $T_i \cup T_j$ to be the terminals of H
3. Define the family Γ of graphs recursively as follows:
   (i) Any $B_i \in B$ is in Γ.
   (ii) If G_i and G_j are in Γ and $\circ_r$ is an operation in O, then the graph $H=G_i \circ_r G_j$ is also in Γ.

Definition 2. Let Γ be a k-terminal recursive family of graphs. The *parse tree* of a graph $G \in \Gamma$ is a tree in which the leaves correspond to the basis graphs from which G is constructed, and each internal node represents the result of applying a composition operation to the graphs represented by the subtrees rooted at its children.

Example 1. The family of *rooted trees* is a 1-terminal recursive family of graphs defined as follows.

1. The triple $(\{x\}, \{\}, \{x\})$ is a rooted tree with root (terminal) x.
2. If $T_1=(V_1,E_1,\{r_1\})$ and $T_2=(V_2,E_2,\{r_2\})$ are rooted trees, then
   $T=(V_1 \cup V_2, E_1 \cup E_2 \cup \{(r_1,r_2)\}, \{r_1\})$ is a rooted tree.



Tree T_1 Tree T_2 Tree $T_1 \circ T_2$ Parse Tree of T_1

Figure 1. Rooted Tree Composition Operation

Example 2. *ProtoHalin graphs* are a 3-terminal recursive family defined as follows.

1. The graph $G = (\{v\}, \{\}, \{v\})$ is a protoHalin graph. v is the root and is also the left and right terminals.
2. If $G_1 = (V_1, E_1, \{t_1, l_1, r_1\})$ and $G_2 = (V_2, E_2, \{t_2, l_2, r_2\})$ are protoHalin graphs, then
   (i) If $G_1 = (\{v\}, \{\}, \{v\})$, then $G = (\{v\} \cup V_2, E_2 \cup \{(v, t_2)\}, \{v, l_2, r_2\})$ is a protoHalin graph.
   (ii) If $G_1 \neq (\{v\}, \{\}, \{v\})$, then $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{(t_1, t_2), (r_1, l_2)\}, \{t_1, l_1, r_2\})$ is a protoHalin graph.

Regular properties are defined in [2] in terms of homomorphisms. Let Γ be a k-terminal recursive family and let $\circ$ be a rule of composition of Γ. Suppose that P is a computable predicate defined on graph × subgraph pairs (G, H) where $G \in \Gamma$ and H is a subgraph of G. In this paper we will only look at subgraphs that are vertex subsets. In this restricted case, we can unambiguously extend the definition of $\circ$ to graph × subgraph pairs by defining $(G_1, H_1) \circ (G_2, H_2) = (G_1 \circ G_2, H_1 \cup H_2)$. A property P is *regular* if, for each composition operator $\circ$, the class of graph × subgraph pairs can be partitioned into a finite number of equivalence classes such that the class to which a pair $(G, H) = (G_1, H_1) \circ (G_2, H_2)$ belongs is a binary function of the classes to which (G_1, H_1) and (G_2, H_2) belong, and for each equivalence class, either all or none of the members of the class satisfy P. Classes which satisfy P are *accepting classes* and classes which do not satisfy P are *rejecting classes*. Generalizations to cases where the subgraph is not a vertex subset are discussed in [2] and [5]. More formally, regularity is defined as follows.

Definition 3. Let Γ be a k-terminal recursive family and let Γ_H be the class of all *graph × subgraph pairs* (G, H) where $G \in \Gamma$ and H is a subgraph of G. Suppose that $\{\circ_1, \circ_2, \cdots, \circ_m\}$, is the set of composition operations of Γ extended to graph × subgraph pairs such that $(G_1, H_1) \circ_r (G_2, H_2)$ is defined whenever $G_1 \circ_r G_2$ is defined. Let T be a set on which m binary operations $\bullet_1, \bullet_2, \cdots, \bullet_m$ are defined (each $\bullet_r$ corresponds to $\circ_r$). Let P be a property (i.e., computable predicate) defined on graph × subgraph pairs. A mapping h from Γ_H onto T is a *homomorphism* with respect to the class Γ and the property P if, for every (G_1, H_1) and (G_2, H_2) in Γ_H, and for each operation $\bullet_r$ corresponding to a composition operation $\circ_r$,

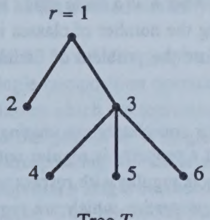
1. $h(G_1, H_1) = h(G_2, H_2)$ implies $P(G_1, H_1) = P(G_2, H_2)$.
2. $h((G_1, H_1) \circ_r (G_2, H_2)) = h(G_1, H_1) \bullet_r h(G_2, H_2)$.

P is *regular* with respect to Γ if it has a homomorphism with a finite range.

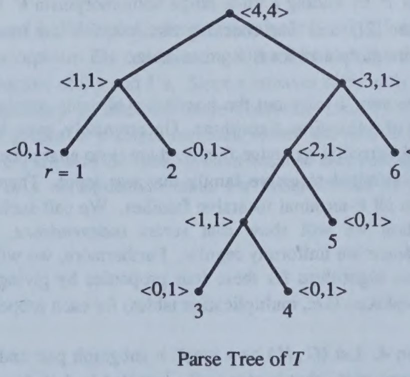
Since the number of equivalence classes is finite for a regular property, each binary operator $\bullet_i$ can be represented as a multiplication table on the set of equivalence classes. Bern, Lawler, and Wong [2] describe a dynamic programming algorithm which uses the multiplication tables for a property and the parse tree of a graph to solve optimal subgraph problems. Starting with the leaves of the parse tree, the algorithm associates an *optimal representative* with each equivalence class at each node of the parse tree. Suppose that v is a node in the parse tree and that G_v is the subgraph that it represents. The optimal representative for an equivalence class at v will be the graph × subgraph pair (G_v, H) in the class for which H has optimum cardinality (or weight). At a leaf, G_v will be a basis graph B_i, and each optimal representative will be a pair of the form (B_i, H). At an interior node of the parse tree, the optimal representatives are functions of the optimal representatives of the children of v. For each equivalence class C_k consider all triples C_i, C_j, $\bullet_l$ such that $C_k = C_i \bullet_l C_j$, and choose the "best" such triple. The optimal solution is the best of the optimal representatives of the accepting classes at the root of the parse tree. Backtracking can be used to construct the optimal subgraph [2]. If only the value of the optimal solution is needed, a vector of length m is stored at each node, where m is the number of equivalence classes. The i^{th} entry in the vector for node v is the optimal value of a subgraph H satisfying P such that (G_v, H) belongs to class C_i. So, the vector at v is related to the corresponding vectors at its children by certain recurrence relations.

It is easily seen that this algorithm is linear in the *size* (number of nodes) of the parse tree. Furthermore, the size of the parse tree is linear in the size of input for any k-terminal recursive family of graphs. Therefore, if the parse tree for any graph in a particular k-terminal recursive family of graphs can be obtained in linear time, then there is a linear time algorithm for finding optimal subgraphs satisfying any given regular property P with respect to that k-terminal recursive family of graphs. The following example illustrates this algorithm.

Example 3. Figure 2 shows a rooted tree, its parse tree, and a multiplication table for independence in rooted trees from [2]. In the multiplication table, C_1 represents the class of all graph × subgraph pairs (G, H) such that H is independent in G and the root of G is not in H, C_2 is the class of all pairs (G, H) such that H is independent in G and the root of G is in H, and C_3 is the class of all pairs (G, H) such that H is *not* independent in G. C_1 and C_2 are *initial classes,* that is, the classes characterizing graph × subgraph pairs on the basis graph. In the parse tree, a vector of length 2 corresponding to classes C_1 and C_2 is stored at each node. Information about C_3 is not stored because it is both a rejecting class and a *sink* class. (A sink class, when composed with any other class yields the same sink class.) Let v_l and v_r denote the left and right children of node v, and let $a_i(v)$ denote the maximum cardinality of an independent set belonging to class C_i at node v. The multiplication table yields the two recurrence relations $a_1(v) = \max\{a_1(v_l) + a_1(v_r), a_1(v_l) + a_2(v_r)\}$ and $a_2(v) = a_2(v_l) + a_1(v_r)$ which relate the vector at each internal node to the vectors at its children. Since C_1 and C_2, are both accepting classes, the maximum cardinality of an independent set is the maximum of the two values at the root of the parse tree (which are both 4 in this example).



Tree T



Parse Tree of T

•	C_1	C_2	C_3
C_1	C_1	C_1	C_3
C_2	C_2	C_3	C_3
C_3	C_3	C_3	C_3

Multiplication Table

Figure 2. Finding Maximum Independent Sets in Rooted Trees

3. PROPERTIES THAT ARE UNIFORMLY REGULAR

The theory described in the previous section guarantees a linear-time algorithm for finding optimal subgraphs whenever the family of graphs is k-terminal recursive, the input consists of a parse tree of a graph, and the property is regular with respect to the family of graphs. A disadvantage of this theory is that it is not constructive; each linear-time algorithm must be constructed "by hand". The theory would be more useful if the construction of the linear-time algorithms could be automated. Ideally, we would like to have an algorithm which, given descriptions of a property, and of a k-terminal recursive family of graphs, produces a linear-time algorithm for finding an optimal subgraph obeying the property for any graph in the family. Unfortunately, this is impossible. Theorem 1 below shows that the problem of determining whether a given property is regular with respect to a given k-terminal recursive family of graphs is (recursively) unsolvable.

Theorem 1. Given a fixed k-terminal recursive family of graphs Γ, and an arbitrary computable property P on graph × subgraph pairs, the problem of finding a finite range homomorphism for P is unsolvable.

Proof. It is known that the problem of determining whether $L = \Sigma^*$ for an arbitrary recursive language L over an arbitrary finite alphabet Σ is undecidable [3]. Since Σ^* is a countable set, we can use diagonalization to map each string in Σ^* to a unique string over an alphabet with one symbol and there is clearly a recursive algorithm to compute this mapping. We can therefore assume that $\Sigma = \{a\}$. Furthermore, since L is recursive, there must be an algorithm (which halts on all inputs) accepting L, so we can assume that L is given in terms of an algorithm A. Consider a particular instance L of this problem and define P to be the property of graph × subgraph pairs on a given k-terminal recursive family of graphs Γ that is true for all graph × subgraph pairs on n vertices exactly when $a^n \in L$, and false otherwise. (Without loss of generality, assume that there exists a graph on n vertices which belongs to the family for all n.) Let A' be an algorithm which inputs graph × subgraph pairs on the given k-terminal recursive family and accepts the pair iff P is true. Clearly P is true for all graph × subgraph pairs on the given k-terminal recursive family iff $L = \Sigma^*$. Furthermore, the homomorphism h with one equivalence class which is an accepting class is a finite range homomorphism for P. Since the problem of determining whether $L = \Sigma^*$ is undecidable, the problem of determining whether h is a homomorphism for a given property P for graph × subgraph pairs on a given k-terminal recursive family is also undecidable. Clearly the problem remains undecidable if h has more than one equivalence class. The problem in the statement of the theorem is to find a finite range homomorphism for P, not to determine whether a given h is finite range homomorphism for P. If the problem of finding a finite range homomorphism for P is solvable, then we could determine whether h is a finite range homomorphism for P by finding a finite range homomorphism h' for P, minimizing the number of classes in both h' and h (see [2]), and then checking that h and h' are isomorphic. Therefore the problem of finding a finite range homomorphism for P is unsolvable. ☐

Theorem 1 rules out the possibility of using regularity as a basis for completely automating the construction of linear-time algorithms. Unfortunately, even if it is known that a property is regular with respect to some k-terminal recursive family, there is no guarantee that the property is regular with respect to the particular k-terminal recursive family that was input. There are, however, properties which are regular with respect to all k-terminal recursive families. We call such properties *uniformly regular*. In the remainder of this section we will show that *vertex independence*, *vertex domination*, *irredundance*, and *vertex j-independence* are uniformly regular. Furthermore, we will show how to partly automate the construction of linear-time algorithms for these four properties by giving effective procedures for constructing finite range homomorphisms (i.e., multiplication tables) for each property for any k-terminal recursive family.

Definition 4. Let (G, H) be a graph × subgraph pair and let V_G and V_H be the vertex sets of G and H. A vertex in V_G that is also in V_H is called *included*; a vertex in $V_G - V_H$ is *excluded*. An *internal vertex* in a graph is a vertex that is not a terminal. (G, H) *satisfies* property P if H satisfies P in G.

Definition 5. A vertex in a graph × subgraph pair is *independent* if none of its neighbours is included. A vertex that is not independent is *dependent*. A graph × subgraph pair (G, H) is *independent* if all its included vertices are independent and the pair is *dependent* otherwise.

Theorem 2. Vertex independence is a uniformly regular property.

Proof. Let Γ be a k-terminal recursive family of graphs. Assume, for now, that all of the composition operations of Γ are defined in the same way for every pair of graphs. If (G, H) is a dependent graph × subgraph pair, then its composition with any other graph × subgraph pair will give a dependent pair. Therefore, all dependent graph × subgraph pairs can be grouped into a rejecting class called REJ. If (G, H) is independent, then any included, independent, internal vertex in (G, H) will remain independent after any composition involving the pair because only the terminals are directly involved in a composition. Therefore, to determine whether the result of a composition operation is an independent graph × subgraph pair or a dependent pair, it is sufficient to know whether both pairs involved in the composition are independent and which of their terminals are included. There are at most 2^k possible ways of including or not including the k terminals of a graph × subgraph pair, so there are at most 2^k accepting classes. The total number of equivalence classes in a finite range homomorphism for independence for Γ is therefore $2^k + 1$ and independence is regular for Γ. Any class containing a pair (G, H) where G is a basis graph is an initial class. For some classes of k-terminal recursive graphs, the composition operations may be defined differently for some pairs of graphs than for others. In particular, some basis graphs in the class may have fewer than k vertices, so they cannot have k distinct terminals. Since the number of graphs for which this can happen is finite, all such differences in the definition of the composition operations can be handled by the addition of a finite number of equivalence classes to the homomorphism. Cases in which the result of the composition of two graphs is undefined can be handled by the addition of an extra rejecting state. Therefore independence is a regular property for any k-terminal recursive family. ☐

To effectively build the multiplication tables for independence for a particular k-terminal recursive family Γ, let T be a set with 2^k accepting equivalence classes and one rejecting class. Each accepting class can be represented by a length k vector of 0's and 1's. A 1 in the i^{th} position indicates that the i^{th} terminal is included and a 0 indicates that it is excluded. The multiplication table for each operation $\bullet_l$ (corresponding to the composition operation $\circ_l$ of Γ can now be genarated by examining the vectors for each pair of classes in T. If C_i and C_j are both accepting classes, and the operation $C_i \circ_l C_j$ does not added an edge between any pair of included terminals of C_i and C_j, then the result is an accepting class whose set of terminals is determined by the definition of $\circ_l$. In all other cases, the result of a composition operation is the rejecting class. The initial classes are the classes containing graph × subgraph pairs (G, H) in which G is a basis graph. The entries in multiplication tables for equivalence classes corresponding to basis graphs with fewer than k vertices can be determined similarly.

Example 4. ProtoHalin graphs, which were defined in Example 2, are a 3-terminal recursive family with a single composition operation $\circ$. The homomorphism has one rejecting class REJ and $2^3 = 8$ accepting classes which are represented by length 3 vectors of 0's and 1's. Since $\circ$ behaves differently when one of the graphs is a single vertex, it is necessary to add two more accepting classes, C_1 and C_2 corresponding the cases where one of the graphs is a single included vertex and where one of the graphs is a single excluded vertex. C_1 and C_2 are also the initial classes because they contain the basis graph. The homomorphism, in the form of a multiplication table, for independence in protoHalin graphs is shown in Figure 3.

•	C_1	C_2	000	001	010	011	100	101	110	111	REJ
C_1	000	011	000	001	010	011	000	001	010	011	REJ
C_2	100	111	100	101	110	111	REJ	REJ	REJ	REJ	REJ
000	000	001	000	001	000	001	000	001	000	001	REJ
001	000	REJ	000	001	REJ	REJ	000	001	REJ	REJ	REJ
010	010	011	010	011	010	011	010	011	010	011	REJ
011	010	REJ	010	011	REJ	REJ	010	011	REJ	REJ	REJ
100	100	REJ	100	101	100	101	REJ	REJ	REJ	REJ	REJ
101	100	REJ	100	101	REJ	REJ	REJ	REJ	REJ	REJ	REJ
110	110	REJ	110	111	110	111	REJ	REJ	REJ	REJ	REJ
111	110	REJ	110	111	REJ	REJ	REJ	REJ	REJ	REJ	REJ
REJ	REJ	REJ	REJ	REJ	REJ	REJ	REJ	REJ	REJ	REJ	REJ

Figure 3. Multiplication Table for Independence in protoHalin Graphs

Definition 6. An included vertex in a graph × subgraph pair *dominates* itself and all of its neighbours. A graph × subgraph pair is dominated if all of its vertices are dominated and is undominated otherwise.

Theorem 3. Vertex domination is a uniformly regular property.

The proof of this theorem is similar to the proof that vertex independence is uniformly regular. The main difference is that three types of terminals must be distinguished: included (and therefore dominated), excluded and dominated, and excluded and undominated. The result is $3^k + 1$ equivalence classes of which 2^k are accepting classes and $3^k - 2^k + 1$ are rejecting classes. When constructing the multiplication tables, each class is represented by a length k vector of 0's, 1's and 2's. More details are contained in [6].

Definition 7. An included vertex in a graph × subgraph pair is *redundant* if the set of vertices that it dominates can be dominated by other included vertices. A vertex which is not redundant is *irredundant*. A subset of vertices S in a graph is *irredundant* if no proper subset of S dominates the same set of vertices as S. An *immune* vertex in a graph is an internal vertex all of whose neighbours are internal vertices. An internal vertex that is not immune is *vulnerable*.

Theorem 4. Irredundance is a uniformly regular property.

The proof of this theorem is more complicated (and longer) than the proofs for independence and domination, but the general idea is the same. In this case it is necessary to examine all vertices within distance two of a particular vertex to determine whether it is irredundant. Three types of excluded terminals and three types of included terminals must be distinguished. Furthermore, for one of the types of included terminals it is necessary to determine the relationship of such a terminal with each other terminal. This results in $2^k + 4$ types of vertices and $(2^k + 4)^k + 1$ equivalence classes. See [6] for details.

Definition 8. Two vertices in a graph are j*-independent*, for $j \geq 2$, if the shortest path between them has at least j edges, and j*-dependent* otherwise. A set of vertices is j-independent if the vertices in the set are pairwise j-independent, and the set is j-dependent otherwise. A graph × subgraph pair is j-independent if its set of included vertices is j-independent. A vertex in a graph × subgraph pair is j-independent if it is j-independent with respect to the set of included vertices. Ordinary independence is 2-independence.

Theorem 5. j-independence is a uniformly regular property for any $j \geq 2$.

Proof. A graph × subgraph pair that is j-dependent can never become j-independent when involved in a composition operation, so all graph × subgraph pairs that are j-dependent can be grouped into a single rejecting state REJ. Now consider j-independent graph × subgraph pairs. The only j-independent vertices in such a pair that could be j-dependent after a composition operation are included vertices that are within distance (i.e., number of edges) $j-2$ of an excluded terminal. To determine the equivalence class of a pair resulting from a composition operation it is sufficient to know the distances of the included vertices from each terminal and the distances between pairs of excluded terminals. For each excluded terminal v, record the set of terminals which are distance i from v for each $1 \le i \le j-2$. In addition, if there is an included internal vertex within distance $j-2$ of v, record the minimum distance of such a vertex. (Otherwise record that there is no such vertex.) This gives at most $(j-1) * (j-2) * 2^{(k-1)}$ possible "states" for each excluded terminal. For each included terminal v, it is only necessary to record the set of excluded terminals which are distance i from v for each $1 \le i \le j-2$. This gives at most $(j-2) * 2^{(k-1)}$ possible states for each included terminal. It is also necessary to indicate which terminals are included and which are excluded. The total number of possible states for a terminal is therefore at most $(j-1) * (j-2) * 2^k$. There are k terminals and there is one rejecting state REJ, so the total number of equivalence classes in a homomorphism for j-independence in any k-terminal recursive family of graphs is at most $((j-1) * (j-2) * 2^k)^k + 1$. Since the number of classes is finite, j-independence is regular for any k-terminal recursive family of graphs. The construction of the multiplication tables for vertex j-independence is similar to the construction of the tables for independence, domination, and irredundance. ☐

4. LOCAL PROPERTIES AND THEIR REGULARITY

We showed in the last section that certain properties are regular with respect to any k-terminal recursive family. All these properties are *local* in the sense that it can be determined whether a subgraph satisfies the property in a graph by checking, for each vertex of the graph, which of the vertices in a bounded neighbourhood are included. For example, a graph × subgraph pair is independent if all of its included vertices are independent, and an included vertex v is independent if none of the neighbours of v are included. So, to determine if a graph × subgraph pair is independent, it is only necessary to examine the 1-neighbourhood (i.e., the immediate neighbourhood) of each vertex and we call independence a *1-local* property. Similarly, a graph × subgraph pair is dominating if all of its vertices are dominated. Since an included vertex is always dominated, and an excluded vertex is dominated if there is an included vertex in its 1-neighbourhood, dominance is also a 1-local property.

Irredundance is an example of a 2-local property. A graph × subgraph pair is irredundant if all of its included vertices are irredundant. An included vertex v in a graph × subgraph pair is irredundant if there is an excluded vertex w in the neighbourhood of v such that the only vertex that is included in the neighbourhood of w is v. Therefore, to check if an included vertex is irredundant we need to check the *2-neighbourhood* of v.

Definition 9. Let (G, H) be a graph × subgraph pair with $|V_G| = n$, and let P be a subgraph property defined on graph × subgraph pairs. Let $N(v)$ denote the *1-neighbourhood* of v (set of vertices adjacent to v), $inc(v)$ the number of included vertices in $N(v)$, and $exc(v)$ the number of excluded vertices in $N(v)$. A *vertex property* p is *compatible* with P if, for all pairs (G, H), $P(G, H)$ is true iff $p(v)$ is true for all vertices $v \in V_G$. Let $[l, u]$ denote an *interval* of non-negative integers (i.e., $[l, u] = \{i \mid 0 \le l \le i \le u \text{ and } i \in Z\}$). A vertex property p is *1-local* if there are four finite sets of disjoint intervals I_1, I_2, I_3, and I_4 such that, for every graph × subgraph pair (G, H) and every vertex $v \in V_G$, $p(v)$ is true iff either v is included, $inc(v) \in I_1$, and $exc(v) \in I_2$, or v is excluded, $inc(v) \in I_3$, and $exc(v) \in I_4$. A subgraph property P is *1-local* if it has a compatible 1-local vertex property.

Example 5. The vertex independence and vertex domination properties are both 1-local. The sets of intervals for the vertex independence property are $I_1 = \{[0, 0]\}$ and $I_2 = I_3 = I_4 = \{[0, \infty]\}$. The sets of intervals for the vertex domination property are $I_3 = \{[1, \infty]\}$ and $I_1 = I_2 = I_4 = \{[0, \infty]\}$.

Theorem 6. All 1-local subgraph properties are regular with respect to all k-terminal recursive families of graphs.

Proof. Let P be a 1-local subgraph property, let p be a vertex property that is compatible with P, and let I_1, I_2, I_3, and I_4 be the sets of intervals associated with p. Let the least upper bound of I_1 be u_1. If the least upper bound of I_1 is ∞, redefine u_1 to be the lower bound of the infinite interval of I_1. Define u_2, u_3, and u_4 similarly. Let (G, H) be a graph × subgraph pair such that the graph belongs to a given k-terminal recursive family of graphs. All internal vertices in the pair that satisfy p will continue to satisfy p when the pair is involved in a composition operation because internal vertices cannot acquire new neighbours. Similarly, if an internal vertex does not satisfy p then it cannot satisfy p after a composition operation. Since P is satisfied for a pair only if all vertices in the pair satisfy p, all pairs that have at least one internal vertex not satisfying p can be grouped into a rejecting class REJ. Now, suppose that all internal vertices in the pair satisfy p. The state of each vertex v can be stored as a pair $(inc(v), exc(v))$. p is true for an included vertex v only if $inc(v) \in I_1$ and $exc(v) \in I_2$. If $inc(v) > u_1$ or $exc(v) > u_2$, then the truth value of $p(v)$ cannot change as a result of more composition operations. So, there are at most $(u_1 + 2)(u_2 + 2)$ possible states for each included vertex. Similarly, there are at most $(u_3 + 2)(u_4 + 2)$ possible states for each excluded vertex. This gives a total of at most $(u_1 + 2)(u_2 + 2) + (u_3 + 2)(u_4 + 2)$ possible states for each vertex. The value of $p(v)$ for an internal vertex v cannot change as a result of a composition operation because v cannot acquire new neighbours. So, to determine whether P is true for a graph × subgraph pair, it is only necessary to store the states of the k terminals and whether p is true for all internal vertices. This gives a total of at most $((u_1 + 2)(u_2 + 2) + (u_3 + 2)(u_4 + 2))^k + 1$ equivalence classes in a homomorphism for P. Since P has a finite range homomorphism, P is regular. The construction of the multiplication tables for P is similar to the constructions in the last section for indepedence and domination. ☐

Definition 10. A vertex property p is j-local for some integer $j \geq 2$ if there is a $(j-1)$-local vertex property α and two sets of disjoint intervals I_1 and I_2 such that, for all graph × subgraph pairs (G, H) and all vertices $v \in V_G$, $p(v)$ is true iff either v is included and the number of vertices in $N(v)$ satisfying α belongs to I_1, or v is excluded and the number of vertices in $N(v)$ satisfying α belongs to I_2. A subgraph property P is j-local if it has a compatible j-local vertex property.

Theorem 7. All 2-local subgraph properties are regular with respect to all k-terminal recursive families of graphs.

The proof of this theorem is a generalization of the proof that 1-local properties are regular. The main difference is that there are more cases to examine because it is necessary to examine the 2-neighbourhood of each vertex. See [6] for details.

Conjecture. For any fixed $j \geq 1$, all j-local properties are regular with respect to all k-terminal recursive families of graphs.

REFERENCES

[1] M.W. Bern, E.L. Lawler, A.L. Wong. "Why Certain Subgraph Computations Require Only Linear Time." *Proc. 26th Symposium on Foundations Of Computer Science*, 1985, pp. 117-125.

[2] M.W. Bern, E.L. Lawler, A.L. Wong. "Linear-Time Computation of Optimal Subgraphs of Decomposable Graphs." *Journal of Algorithms 8:2 (June 1987)*, pp. 216-235.

[3] J.E. Hopcroft, J.D. Ullman. "Introduction to Automata Theory, Languages, and Computation." Addison-Wesley, 1979, Chap. 8.

[4] E.L. Lawler, P.J. Slater. "A Linear Time Algorithm for Finding an Optimal Dominating Subforest of a Tree." Manuscript, July 1985.

[5] S. Mahajan. "Sequential and Parallel Optimization Algorithms for Recursive Graphs." *M.Sc. Thesis, School of Computing Science, Simon Fraser University*, December 1986.

[6] S. Mahajan, J.G. Peters. "Algorithms for Regular Properties in Recursive Graphs." *Technical Report TR 87-7, School of Computing Science, Simon Fraser Univ.*, 1987.

[7] J. Pfaff, R. Laskar, S.T. Hedetniemi. "Linear Algorithms for Independent Domination and Total Domination in Series-Parallel Graphs." *Congressus Numerantium* 45, 1984, pp. 71-82.

[8] K. Takamizawa, T. Nishizeki, N. Saito. "Linear-Time Computability of Combinatorial Problems on Series-Parallel Graphs." *Journal ACM 29:3 (July 1982)*, pp. 623-641.

[9] T.V. Wimer. "Linear Algorithms on k-Terminal Recursive Graphs." *Ph.D. Thesis, Dept. of Computer Science, Clemson University*, 1987.

[10] T.V. Wimer, S.T. Hedetniemi. "K-terminal Recursive Families of Graphs." *Proc. 250th Anniversary Conf. on Graph Theory (Fort Wayne, Ind., 1986). Utilitas Mathematica*, to appear.

[11] T.V. Wimer, S.T. Hedetniemi, R. Laskar. "A Methodology For Constructing Linear Graph Algorithms." *Congressus Numerantium* 50, 1985, pp. 43-60.

[12] T.V. Wimer, K. Peters, R. Laskar, S.T. Hedetniemi, E.O. Hare. "Linear Time Computability of Combinatorial Problems on Generalized Series-Parallel Graphs." *Perspectives in Computing, Vol. 15*, Academic Press, 1987, pp. 437-457.

An Optimum $\Theta(n \log n)$ Algorithm for finding a Canonical Hamiltonian Path and a Canonical Hamiltonian Circuit in a set of intervals.

Glenn K. Manacher and Terrance A. Mankus
Department of Mathematics, Statistics and Computer Science
University of Illinois at Chicago
Chicago, Illinois 60608

The problem of finding a Hamiltonian Path (HP) or a Hamiltonian Circuit (HC) in polynomial time for an interval graph or its model (an explicit set of intervals) has been examined before. Bertossi [B 85] examined the case of proper intervals, in which no interval may contain another. In this case the HP problem is trivial since a HP exists if and only if the graph is connected; a left-to-right left-endpoint ordering then gives a HP. From [FW 78] Bertossi deduced that, if the intervals are presented as real-number pairs, a lower bound $\Omega(n \log n)$ must hold. The main contribution of [B 83] was to show that the existence and construction of a HC could be achieved in time $\Theta(n \log n)$ for proper intervals.

It is well known that the existence problem for HP and HC in general graphs is NP-complete.

The problem of finding a HC for a general interval model (with no restriction that it be proper) in polynomial time was solved by Keil [K 85]. He showed that this problem could be solved in time and space $\Theta(n+e)$ where e is the number of edges in the interval graph. This was done by using an explicit sequence of maximum cliques, which is easily shown to require space $\Theta(n+e)$. His algorithm has the same complexity as Booth and Lueker's interval graph recognition algorithm [BL 76] and so is optimal in the size of the underlying graph. In fact, [K 85] can be modified, by the use of a different data structure, to run in space $\Theta(n)$ and time $\Theta(n \log n)$ but it does not have the simplicity of our algorithm nor does it produce a natural canonical HP or HC if there is one.

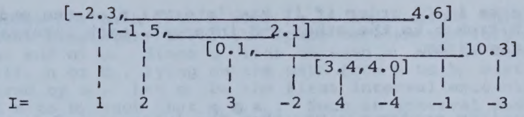
We show that there exists a natural canonical form for a HP and a HC that may be produced by an extremely simple algorithm in optimal time and space. Both of these algorithms assume a presentation in the form of an unsorted list of intervals consisting of pairs of reals. If the presentation is sorted, the time complexity can be reduced to $O(n \log\log n)$ by using well-known techniques of [vEB 77].

BASIC DEFINITIONS. An interval is a pair of real numbers (x,y), $x < y$, denoting a line segment encompassing all points p in R^1 for which $x \leq p \leq y$. Given a set I of such intervals, it is possible to sort them in left endpoint order. We will label (or index) an interval by the position of its left endpoint. Thus the leftmost left endpoint is labeled or indexed 1. The rightmost left endpoint will be labeled n, if I has n intervals. This sorting and labeling can be done in time $O(n \log n)$. Two intervals intersect iff they share a point if common in R^1. If two intervals do not intersect, then the interval with smaller index lies completely to the left of the interval with larger index.

An intersection graph (hereafter, graph) of a set of intervals is a graph whose vertices represent the intervals in the set and whose edges represent intersection between two vertices. If a graph can be built up in this way from a set of intervals it is called an interval graph, and the set of intervals, presented as above, is called its interval model or simply model.

The listing I is the interval model of the set of real intervals. S will denote the set $\{1,2,\ldots,n\}$ if the interval model I has n intervals. We will use the left endpoint index or label when referring to the corresponding interval in the interval set.

Figure 1.



Let A and B be two distinct intervals in some interval model. We will use the following notations:

$A \oplus B$ means "A intersects B",

$A \not\oplus B$ means "A is disjoint from B",

$A < B$ means "The left end point of A lies to the left of the left end point of B". Alternately, "The label of A is less than the label of B".

A <u>path</u> is a listing of indices, such that any two indices listed in immediate succession indicates the corresponding intervals intersect in the real line.

A <u>Hamiltonian path</u> (if one exists) is a listing of all indices in S, each appearing once. It will be denoted by <u>HP</u>.

Path notations: $A\_\_B$ means that A is directly connected to B, with no intervening intervals on the path from A to B.

$A\overset{0}{\_\_}B$ means that a path connects A and B and there may be 0 or more intervals on the path from A to B.

$A\overset{1}{\_\_}B$ means that a path connects A and B and there is at least 1 interval on the path from A to B.

A... means that a path from A may exist, but it is not crucial to the completion of the current proof.

Example: A HP for the above example is $4\_\_1\_\_3\_\_2$.

When drawing an interval, ____ _ _ _ , _ _ _ ____ or _ _ ____ _ _ describes an interval that may or may not extend along the dotted lines, but if it does, those portions of the interval will not matter to the proof.

The <u>end intervals</u> are the first and last indices of a path listing. In the above HP listing, the end intervals are 4 and 2.

A <u>Hamiltonian circuit</u> (if one exists) is a HP in which the end intervals intersect. It will be denoted by <u>HC</u>.

From now on, let I be an interval model with n intervals, and let S be the set of indices of I.

<u>CS/HP ALGORITHM FOR HP PRODUCTION:</u>*

Input: Ordered listing of the intervals in I (as per Figure 1);

Output: Ordered listing of intervals in HP or Failure;

```
1) Set $S_1 \leftarrow \{1,2,...,n-1\}$ ; Set LAST $\leftarrow$ n; Set $p_1 \leftarrow n$;
2) Set PATH $\leftarrow p_1$ ;
3) FOR i=2 TO n DO;
4)    Find largest interval in $S_i$ that intersects LAST, if one exists.  If one
      exists, set $p_i$ to that index.  Otherwise report Failure and Stop;
5)    Set PATH $\leftarrow$ PATH$\_\_p_i$ ; Set LAST $\leftarrow p_i$ ; Set $S_{i+1} \leftarrow S_i - \{p_i\}$ ;
6) END DO;
7) REPORT PATH=$p_1\_\_p_2\_\_p_3 \ldots p_n$ ;
```

<u>MAIN HP THEOREM</u> The CS/HP algorithm produces a HP for I if and only if I has a HP.

*CS refers to Ms. Carol Smith, who discovered this ordering and first brought these matters to our attention.

A HP will be said to be in <u>CS order</u> if it has interval n as one end interval, and on the path from n to the other end interval, each interval is the choice of the CS/HP algorithm.

Proof (only if)

Trivially, if the CS algorithm produces a HP, then the interval model must have one.

(if)
To prove the remainder of the theorem, we need a lemma.

<u>Lemma 1</u>. If I is an interval model that has a HP, then I has a HP with interval n as one end interval.

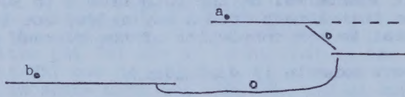
<u>Note:</u> All our path transformations preserve "Hamiltonicity" of paths.

Proof of Lemma 1:

Let T be any HP of I. If either end interval is n then we are done. So we suppose that n is not an end interval. Let a_0 and b_0 be the two end intervals of T. Then T= $a_0 \overset{\circ}{\_\_} n \overset{\circ}{\_\_} b_0$. We now have two cases.

Case 1) $b_0 \oplus a_0$. We can trivially realign T as T'= $n \overset{\circ}{\_\_} b_0 \_\_ a_0 \cdots$ or T'= $n \overset{\circ}{\_\_} a_0 \_\_ b_0 \cdots$ by disconnecting the link at n and joining a_0 and b_0 .

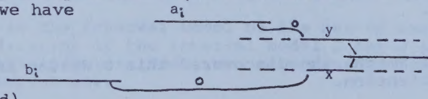
Case 2) $b_0 \not\oplus a_0$. Suppose $b_0 < a_0$. Then b_0 lies completely to the left of a_0 . Pictorially we have



Note: The length of (or number of intervals in) a path is finite, since the number of intervals in the model S is finite. Hence, the lengths of $b_0 \_\_ n$ and $a_0 \_\_ n$ are finite. Depending on certains conditions on the end intervals of these "arms", they will either be trivially reconnected or one arm will be "pruned", thus shortening its length, The new end intervals will then be relabeled for another "pruning", this pruning occuring on the other arm. The pruned pieces are reconnected in a precise manner which will result in the formation of the desired HP. The pruning shortens the lengths of the arms in an alternating fashion, and since the arms have finite length, the pruning must enventually stop, at which time the desired HP is finally produced.

We present the following procedure that accomplishes this realignment.

```
PROCEDURE
Initializations: Set $i \leftarrow 0$;  Set PATH $\leftarrow \Lambda$;( the null string )
                 Set $T_i \leftarrow a_0 \overset{\circ}{\_\_} n \overset{\circ}{\_\_} b_0$ ; (The initial HP.)
LOOP;
  IF($a_i \oplus$ n) THEN
    DO;Since $b_i < a_i$ and $b_i \not\oplus a_i$ , we have $b_i \not\oplus n$. So the path can be written as
       $T_i = b_i \overset{\circ}{\_\_} x \_\_ n \overset{\circ}{\_\_} a_i$ , where $x < n$. Since $a_i \oplus n$, $x \oplus n$, $a_i < n$ and $x < n$, this
       means $a_i \oplus x$. Hence we can realign the path as $T_i \leftarrow n \overset{\circ}{\_\_} a_i \_\_ x \overset{\circ}{\_\_} b_i$ and we are
       done. So Set PATH $\leftarrow$ PATH $\_\_ b_i \overset{\circ}{\_\_} a_i \overset{\circ}{\_\_} n$;    EXIT LOOP;
    END DO;
  ELSE
    DO;If we are here, then $a_i \not\oplus n$. Since there is a path $a_i \overset{1}{\_\_} n$, there must
       exist two intervals x and y such that the path $T_i$ is $b_i \overset{\circ}{\_\_} x \_\_ n \_\_ y \overset{\circ}{\_\_} a_i$ with
       $a_i$ lying completely between $b_i$ and n; clearly $x \oplus y$, $x \oplus n$ and $y \oplus n$.
       Pictorially we have
```



(Continued)

```
          Consider the path from n to $b_i$ ,$n \text{---} x \overset{\circ}{\text{---}} b_i$ .  Since this path is connected,
          it must completely cover the space between the left end of $b_i$ and the
          right end of n.  Since $a_i$ lies between $b_i$ and n, some interval, not
          itself, n or $b_i$ , lying on the path from n to $b_i$ must cross the space
          covered by $a_i$ .  Let $q_i$ be the first interval encountered on the path
          from n to $b_i$ such that $q_i \oplus a_i$ .  Such an interval must exist because the
          path is connected.  Let $d_i$ be the interval just before $q_i$ on this path.
          Clearly, $d_i \not\oplus a_i$ .

          IF($d_i$ =n) THEN
             DO;We have that $q_i \oplus n$ and our path $T_i$ is $b_i \overset{\circ}{\text{---}} q_i \text{---} n \text{---} y \overset{\circ}{\text{---}} a_i$ .  Since
                $q_i \oplus a_i$ , we may realign the path $T_i$ as $b_i \overset{\circ}{\text{---}} q_i \text{---} a_i \overset{\circ}{\text{---}} n$.
                Set PATH$\leftarrow$PATH$\text{---} b_i \overset{\circ}{\text{---}} n$; EXIT LOOP;
             END DO;
          ELSE
/* A */      DO;When here, $d_i < n$, $d_i \not\oplus a_i$ and $a_i < d_i$ .  $T_i$ is $b_i \overset{\circ}{\text{---}} q_i \text{---} d_i \overset{\circ}{\text{---}} n \overset{\circ}{\text{---}} a_i$ We
                notice that the path $d_i \overset{\circ}{\text{---}} n$ is "shorter" (has fewer intervals)
                than $b_i \overset{\circ}{\text{---}} q_i \text{---} d_i \overset{\circ}{\text{---}} n$ by the length of the seqment $b_i \overset{\circ}{\text{---}} q_i$ .  So the
                length of the subpath with leftmost end interval will be
                decreased by at least 2 intervals.  We immediately realign our
                path $T_i \leftarrow b_i \overset{\circ}{\text{---}} q_i \text{---} a_i \overset{\circ}{\text{---}} n \overset{\circ}{\text{---}} d_i$ .  We notice by the construction and
                the realignment that $a_i < d_i$ , $a_i \not\oplus d_i$ , $d_i < n$. This is the identical
                setup for entry into the Procedure LOOP under the following
                relabelings: Set $b_{i+1} \leftarrow a_i$ ; Set $a_{i+1} \leftarrow d_i$ .  Thus we are in a position
                to repeat this procedure when returning to the LOOP statement
                after completion of this DO group.  Also, the fact that $b_{i+1} = a_i$
                means that $b_{i+1} \oplus q_i$ .  This permits us to prune the path segment
                $b_i \overset{\circ}{\text{---}} q_i$ and adjoin it to PATH, since PATH= $\ldots q_{i-1}$.
                Set PATH $\leftarrow$PATH$\text{---} b_i \overset{\circ}{\text{---}} q_i$ ; Set $T_{i+1} \leftarrow b_{i+1} \overset{\circ}{\text{---}} n \overset{\circ}{\text{---}} a_{i+1}$;  Set i$\leftarrow$i+1;
              END DO;
      END DO;
END LOOP;
Comment - The EXIT LOOP statements will cause execution to resume here.
Comment - When we get here, we are done.

OUTPUT;  Report that the realigned path is PATH= $b_0 \overset{\circ}{\text{---}} q_0 \text{---} b_1 \overset{\circ}{\text{---}} q_1 - \ldots\ b_i \overset{\circ}{\text{---}} n$;
END Procedure;
```

Our path is now realigned to have n as an end interval.

It is clear that the path arms are alternately decreased in length by at least 2 intervals in the /* A3 */ DO group. There are only a finite number of intervals in each arm and each arm has at least 1 interval that intersects n, otherwise there would be no path. Therefore, the procedure must eventually terminate successfully. Hence Lemma 1 is proved.
(End of Lemma 1 proof)

We now continue with the proof of the MAIN HP theorem. We will use induction on the size of an interval model. We will assume that the theorem is true for any interval model I with $|S| \le n-1$.

The cases with n=1 and n=2 are trivial. We can easily see that there are only 3 case of interest when n=3. In each case, $1 \oplus 2$.
Case 1) $1 \not\oplus 3$, $2 \oplus 3$. This has HP 1-2-3 which is in CS order.
Case 2) $1 \oplus 3$, $2 \not\oplus 3$. This has HP 2-1-3 which is in CS order.
Case 3) $1 \oplus 3$, $2 \oplus 3$. This has HP 1-3-2 which can be realigned to give HP 1-2-3 which is in CS order.
Thus we see the initial step of the induction is true. That is, if a HP exists, it can be put in CS order.

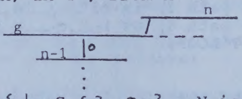
Let I be any interval model of size n, with a HP T. We can use Lemma 1 to realign the path T into a new path T', T'=n..., with n as an end interval. This is the starting point for the CS algorithm.

We will now resort to cases.

case 1. In I, $n-1 \oplus n$. We then take T' and prune n, form the new set I', where $I'=I-\{n\}$, $|I'|=n-1$, and T'' is a HP. By the induction hypothesis, T'' can be realigned in CS ordering starting with n-1. Let this path be T'''. We can then rejoin n to T''' at n-1 to form T^{iv}. The first interval is in correct position, the remaining intervals are in their correct positions and the first interval joins correctly with the next interval in the CS ordering. Thus T^{iv} is the CS ordered HP for the interval model I.

Case 2. In I, $n-1 \not\oplus n$. Since, by assumption, a HP exists, there is at least one interval on the path, in T', from n to n-1. So, $T'=n\_\_g\overset{\circ}{\_\_}n-1\ldots$

Pictorially,



Trivially, we have $g < n-1$,
$g \oplus n$, and
$g \oplus n-1$.

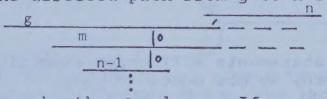
Consider the set $N=\{z \mid z \in S-\{n\}, z \oplus n\}$. N is not empty since $g \in N$. Let m be the interval in N with the largest index (rightmost interval). This is the second interval in the CS ordering. We will show that T' can be realigned with m in the correct position so that the induction step may be used.

case 2a. m=g. So $T'=n\_m\overset{\circ}{\_\_}n-1\ldots$ We can consider $I'=I-\{n,m\}$, and the pruned path $T''=\ldots n-1\ldots$. $|I'|=n-2$, and T'' is the HP. Thus we may use the induction hypothesis to form a CS ordered HP $T'''=n-1\ldots$. We can adjoin $n\_\_m$ to T to create $T^{iv}=n\_\_m\_\_n-1\ldots$ which is in correct CS order (same argument as case 1).

case 2b) $m>g$. We have 2 subcases.

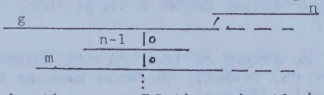
sub 1. $T'=n\_\_g\overset{\circ}{\_\_}m\overset{\circ}{\_\_}n-1\ldots$

m lies on the directed path from g to n-1. Pictorially



Consider the subpath $m\overset{\circ}{\_\_}n-1\ldots$. If $m\_\_n-1\ldots$, we can realign T' as $T''=n\_\_m\_\_g\overset{\circ}{\_\_}n-1\ldots$. If $m\overset{1}{\_\_}n-1\ldots$, then $m\_\_w\overset{\circ}{\_\_}n-1\ldots$, for some $w \in I$. So our path T'' is $n\_g\overset{\circ}{\_\_}m\_\_w\overset{\circ}{\_\_}n-1\ldots$. We may use Lemma 1 on $w\overset{\circ}{\_\_}n-1\ldots$ to make the subpath start with n-1, $n-1\ldots$. We use this realigned subpath to form $T'''=n\_\_m\overset{\circ}{\_\_}g\_\_n-1\ldots$. In either case, we have the path start in correct CS ordering. We again form $I'=I-\{n,m\}$ and repeat the argument of case 2a to obtain our CS ordered HP for I.

sub 2. The path $T'=n\_\_g\overset{\circ}{\_\_}n-1\overset{\circ}{\_\_}m\ldots$. Pictorially



Consider the subpath $m\ldots$. If the subpath is only the interval m, then we trivially realign T' as $T''=n\_\_m\overset{\circ}{\_\_}n-1\overset{\circ}{\_\_}g$ since $m \oplus n$. If the subpath is more than m, it is $m\_\_w\ldots$, for some $w \in I$. We have $w<n-1$, and $w \oplus m$. Since $g<m$, $g \oplus n$ and $m \oplus n$, this means that $w \oplus g$. So we can realign as $T''=n\_\_m\overset{\circ}{\_\_}n-1\overset{\circ}{\_\_}g\_\_w\ldots$. In either case, we have $T''=n\_\_m\overset{\circ}{\_\_}n-1\overset{\circ}{\_\_}g\_\_\ldots$. So we again form $I'=I-\{n,m\}$, $|I'|=n-2$, with T''' pruned of n and m. We again repeat the argument of case 2a to arrive at the correct HP in CS order.

So we see that if the theorem is true for all interval models of size n-1 or less, then it is true for all interval models of size n. It is true for n=1,2,or 3, hence it is true for all interval models.

The MAIN HP THEOREM is thus proved.

MAIN HC THEOREM The CS/HC algorithm produces a HC if and only if the interval model I has a HC. (CS/HC algorithm is in appendix A.)

Proof (if)

Clearly, if the CS algorithm produces a HC, then the interval model I must have a HC.

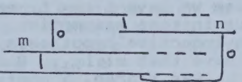
(only if)
We will use induction to show that any HC of I can be realigned (or restructured) so that the intervals in the HC are in proper CS order, and thus would be produced by the CS/HC algorithm.

Suppose I has a HC (Call it T). Then n is present in T. This is the starting point for the CS/HC algorithm. We will restructure T so that at each step, the interval that the CS/HC algorithm chooses will be correctly realigned leaving the HC intact.

Consider the set N from the previous theorem proof. Since interval n has the rightmost left end, there must be two or more intervals in N, otherwise T would not exist. Let $m \in N$ be the largest interval in N. Since the number of intervals is finite and there are at least two in N, this largest interval m must exist. We have two cases.

Case 1. $m \_\_ n$ in the HC. We have the desired starting condition. So let $U_1 = n$, and let $L_1 = m$. We now have the correct arm beginnings for the CS algorithm.

Case 2. $m \overset{1}{\_\_} n$ in the HC. The HC T is of the form $\overset{\circ}{\lfloor n \_\_ y \overset{\circ}{\_\_} m \_\_ x \rfloor}$ where $y < m < n$, $x < n$ and $x \oplus m$. Hence, $x \oplus y$. So we unlink $y \overset{\circ}{\_\_} m$ and reverse the order to $m \overset{\circ}{\_\_} y$. Now $m \oplus n$ and $x \oplus y$ so we can relink to form $\lfloor n \_\_ m \overset{\circ}{\_\_} y \_\_ x \rfloor^{\circ}$ and our HC T has the desired CS starting property, namely that the largest available interval that intersects n is linked directly to n in the HC. So let $U_1 = n$, $L_1 = m$.
Pictorially



At this point we will need 3 lemmas that apply to the conditions at each stage of the CS algorithm.

Let I be an interval model, with interval index set S. Let $U_0 = \infty$ and $L_0 = \infty$. Let $U = \{U_1, U_2, \ldots, U_k\}$, and $L = \{L_1, L_2, \ldots, L_j\}$ be the U arm and L arm of the CS algorithm as constructed so far. U_k is the last placed on the U arm and L_j is the last placed on the L arm. Let $S_{k,j}$ be the set of indices not yet placed in U or L. Let m be the largest index in $S_{k,j}$ ($m = \max(S_{k,j})$). Under these conditions, we then have

LEMMA 2 $\max(S_{k,j}) < \max(U_k, L_j)$,

LEMMA 3 If $\max(S_{k,j}) > \min(U_k, L_j)$, then $\max(S_{k,j}) \oplus \min(U_k, L_j)$, and

LEMMA 4 $w \in S_{k,j}$, $w > \min(U_k, L_j)$, then $w \not\oplus \max(U_k, L_j)$.

Proof of Lemma 2.
Since $U_1 = n$ by the conventions of the CS algorithm, we see that $n-1 = \max(S_{1,0})$ and $n = \min(U_1, L_0)$. If the CS algorithm can continue, then there is a $d \in S_{1,0}$, $d \oplus n$ and d is the largest such index in $S_{1,0}$. The by the rules of the CS/HC algorithm, $L_1 = d$.

If $d = n-1$, then $n-2 = \max(S_{1,1})$ and $\max(S_{1,1}) < \max(U_1, L_1)$. If $d < n-1$, then $n-1 = \max(S_{1,1})$, and still $\max(S_{1,1}) < \max(U_1, L_1)$. In either case, the lemma is true for $S_{1,0}$ and $S_{1,1}$.

We can assume that the CS/HC algorithm procedes correctly until the first time that $\max(S_{k,j}) > \max(U_k, L_j)$.

Suppose $L_j < U_k$. At this stage, denote $\max(S_{k,j})$ by the letter m. We will consider the removal of intervals in the reverse order of placement.

If U_k was the last interval extracted from $S_{k-1,j}$, then $\max(S_{k-1,j})$ was m, since $U_k < m$. By the assumptions of the proof, we must have $m < U_{k-1}$. If $m \oplus U_{k-1}$, it would have been chosen before U_k. Thus $m \not\oplus U_{k-1}$. We can repeat this removal argument with successively removed U_l's, until it would be L_j's turn for removal. Let those removed indices be $U_{k-1}, U_{k-2}, \ldots, U_v$. The assumptions of the proof lead to $L_j < U_k < m < U_{k-1} \ldots < U_v < U_{v-1}$. At this stage, $\max(S_{v-1,j-1})$ is U_v. The placement of L_j before U_v violated the CS ordering. Any first time violation of Lemma 2 during CS/HC construction leads to a similar contradiction. The same type of argument applies if we suppose that L_j was last extracted or if $U_k < L_j$.

Hence Lemma 2 is proved.

Proof of Lemma 3.

The same constructions for the initial steps of Lemma 2 apply to Lemma 3. Again we suppose that at the stage when U_k and L_j are the ends of their arms, Lemma 3 was true for all previous stages of the CS/HC algorithm construction, and that this is the first time during the current construction that $\max(S_{k,j}) > \min(U_k, L_j)$ but that $\max(S_{k,j}) \not\oplus \min(U_k, L_j)$.

Suppose that $L_j < U_k$. By Lemma 2, $m < U_k$.

If U_k was the last placed index, then U_k was $\max(S_{k-1,j})$. By Lemma 2, $U_k < U_{k-1}$By the hypothesis of this proof, since $L_j < U_k$, $U_k \oplus L_j$. We have $L_j < m < U_{k-1}$, and $L_j \oplus U_k$. This means that $m \oplus L$, contradicting our assumption that $m \not\oplus L_j$. So U_k must not have been last placed.

If L_j was last placed during construction, then we have two possibilities.
Case 1) $m < L_{j-1}$. In this case we have $L_j < m < L_{j-1}$ and $L_j \oplus L_{j-1}$. So $L_j \oplus m$. Contradiction to our initial assumption.
Case 2) $L_{j-1} < m$. Then by the inductive hypothesis, since m is $\max(S_{k,j})$ and $m > \min(U_k, L_{j-1})$, we have that $m \oplus L_{j-1}$. But then m should have been placed on L before L_j was placed, a contradiction to the rules of CS construction.

All the above arguments lead to a contradiction if this was the first occurrence of a violation of Lemma 3. But if there is to be any violation, there must be a first time. Hence there must be no violations of Lemma 3. The proof of the case when $U_k < L_j$ proceeds in a similar manner. Lemma 3 is proved.

Proof of Lemma 4.

Again, the initial steps are similar to those in the proof of Lemma 2 and Lemma 3. We assume that there is a first occurrence when $w \in S_{k,j}$ is of larger index than $\min(U_k, L_j)$ and such that $w \oplus \max(U_k, L_j)$. Suppose $L_j < U_k$. By Lemma 2 we must have $L_j < w < U_k$. We also have $w \oplus U_k$ by assumption. L_j could not have been placed last, since ,at least, w was available at the time. U_k must have been placed last. Repeat the removal argument of the proof of Lemma 2 and arrive at a similar contradiction. Our assumption that $w \oplus U_k$ must be wrong, the reasoning being similar to that of the proof above. Hence, Lemma 4 is proved.

We now continue with the proof of the MAIN HC theorem.

We suppose that the HC of I has been successfully realigned in CS order when we reach the stage where U_k is the last placed on the U arm and L_j is the last placed on the L arm, and that a HC is preserved. Let CAND (short for "candidate") be the index chosen by the CS algorithm as next to be adjoined to the U or L arm. If no such CAND exist, then no HC exists. Note that CAND need not be $\max(S_{k,j})$.

Suppose $L_j < U_k$.

Case 1) $\text{CAND} \oplus U_K$. If CAND is directly connected to U_K in our current HC, set $U_{K+1}=\text{CAND}$, and set $S_{K+1,j}=S_{K,j}-\{\text{CAND}\}$, and we proceed. Otherwise, there exist $a \in S_{K,j}$, such that the HC is $L_j \overset{0}{-} \text{CAND} \overset{0}{-} a - U_K$. (Note: in this and all future HC diagrams, we will omit the showing of the path $U_K \overset{0}{-} L_j$). We must have $a < L_j$ by Lemma 4. Also, $a < \text{CAND}$. If CAND is directly connected to L_j, we reverse the path $a \overset{0}{-} \text{CAND}$ and reconnect as $L_j - a \overset{0}{-} \text{CAND} - U_K$, since $a \oplus L_j$ by way of Lemma 4. If CAND is not directly connected to L_j, then there exist a $b \in S_{K,j}$ such that our HC is $L_j \overset{0}{-} b - \text{CAND} \overset{0}{-} a - U_K$. Now $b < U_K$, $b \oplus \text{CAND}$, so $b \oplus a$. So we relink as $L_j \overset{0}{-} b - a \overset{0}{-} \text{CAND} - U_K$, and our HC is intact and in proper CS order.

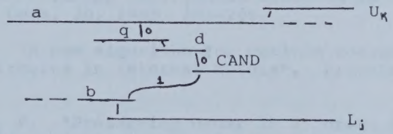
Case 2) $\text{CAND} \not\oplus U_K$. In this case there are two subcases, $\text{CAND} < L_j$ or $L_j < \text{CAND}$. In either subcase, if CAND is directly connected to L_j in the HC, then we are finished with this stage. So suppose that the HC is of the form $L_j - b \overset{0}{-} \text{CAND} \overset{0}{-} a - U_K$, where $a < \text{CAND}$, $a < L_j$ (by Lemma 4).

subcase 1) $\text{CAND} < L_j$. Then $b < \text{CAND} < L$, $b \oplus L$, $a < \text{CAND} < L$ and $a \oplus U_K$, so $a \oplus b$. If CAND is directly connected to a, then our HC is $L_j - b \overset{0}{-} \text{CAND} \overset{0}{-} a - U_K$. But $a \oplus b$, so we can realign our HC as $L_j - \text{CAND} \overset{0}{-} b - a - U_K$, which is in proper CS order. Otherwise, there exists $d \in S_{K,j}$ such that the HC is $L_j - b \overset{0}{-} \text{CAND} - d \overset{0}{-} a - U_K$. Now $b < \text{CAND} < L_j$, $d < L_j$, $d \oplus \text{CAND}$ and $b \oplus L_j$, so $d \oplus b$. Hence we can realign the HC as $L_j - \text{CAND} \overset{0}{-} b - d \overset{0}{-} a - U_K$. This is in proper CS order. Thus, in this subcase, we can realign the HC to the proper next stage and still retain the HC.

subcase 2) $L_j < \text{CAND}$. If CAND is directly connected to L_j, we are done at this stage. Suppose the HC is of the form $L_j - b \overset{0}{-} \text{CAND} \overset{0}{-} a - U_K$, with $b < \text{CAND}$, $b \oplus L_j$ and $a < L_j$.

Let $b \oplus \text{CAND}$, we look at the path segment $a \overset{0}{-} \text{CAND}$. If $a - \text{CAND}$, we relink the HC as $L_j - \text{CAND} \overset{0}{-} b - a - U_K$, since $b \oplus a$. If $a \overset{1}{-} \text{CAND}$, then the path is of the form $a \overset{0}{-} d - \text{CAND}$ for some $d \in S_{K,j}$. Since $L_j < \text{CAND}$, $d \oplus \text{CAND}$ and $b \oplus \text{CAND}$, we see $d < \text{CAND}$. Hence, $d \oplus b$. We relink as $L_j - \text{CAND} \overset{0}{-} b - d \overset{0}{-} a - U_K$.

Let $b \not\oplus \text{CAND}$. We again look at the path $a \overset{0}{-} \text{CAND}$. Since b lies completely to the left of CAND, $L_j < \text{CAND}$ and $\text{CAND} \oplus L_j$, there is a segment of L_j between the right end point of b and the left end of CAND. Since the path from CAND to a is connected, there are intervals q and d, such that on the path from CAND to a, $b < d$, $b \not\oplus d$, $q \oplus b$ and d is directly connected to q.
Pictorially, (note: d may equal CAND and q may be a.)



Our HC is $L_j - b \overset{0}{-} \text{CAND} \overset{0}{-} d - q \overset{0}{-} a - U_K$, with $b \oplus q$, $b < d$ and $b \not\oplus d$. Now CAND can be seen to be $\max(S_{K,j})$ since $L_j < \text{CAND}$, by Lemmas 3 and 4. Consider the subpath $b \overset{1}{-} \text{CAND} \overset{0}{-} d$. Every interval in the path is in $S_{K,j}$, and thus less than CAND. This subpath is connected and so is a HP for some interval set. We are thus is a position to use the proof construction of the HP theorem to realign this subpath into a new subpath $b \overset{1}{-} \text{CAND}$. (See the PROCEDURE for details.) Thus we are able to realign the HC as $L_j - \text{CAND} \overset{0}{-} b - q \overset{0}{-} a - U_K$, and we were able to correctly link CAND to the proper arm.

In all cases when I has a HC, we were able to keep, at each stage of realignment, the CS ordered arms and realign the remaining intervals so that the next interval in the CS ordering was placed correctly, and a HC remained. We showed that the first two intervals could always be aligned properly in CS order. In this manner we have shown that all the remaining intervals can be placed in the proper order while maintaining the HC.

Thus the HC Theorem is proved.

ANALYSIS OF ALGORITHM COMPLEXITY

To see that our HP algorithm works in time $\Theta(n \log n)$, we need only show that "Find the largest interval in S_i that intersects LAST" and "Set $S_{i+1} \leftarrow S_i - \{index\}$" can be done in time $\Theta(\log n)$ using a linear space data structure. It is well known [AHU 74] that one can build an ordered heap in linear time on an already pre-ordered set. In this instance, let the heap have intervals as leaf nodes, ordered on their left endpoint. Each interior node contains a value consisting of the maximum right endpoint of the intervals of the subtree whereof that node is a root. This structure clearly requires linear space. The search for the largest intersecting interval in S_i is accomplished due to the following fact: No interval in S_i can lie completely to the right of LAST.

To accomplish a "Find", we use the left endpoint of the last added interval, LAST, and search the heap in the following manner. If the root value is less than LAST, no interval intersects and a failure results. Otherwise, we traverse the heap looking for the rightmost leaf node whose right endpoint is larger than LAST. This traversal is clearly $\Theta(\log n)$. When the leaf node is found, we remove it from S_i by changing its value to -1, and reversing our traversal, resetting the values of the nodes on the way back to the root. Both these operations are clearly $\Theta(\log n)$ [AHU 74]. Thus our algorithms both need time $\Theta(n \log n)$ and space $\Theta(n)$.

Some final words about the paper.
After much joint effort required to develope various strategies and lemmas, each author succeeded in producing different kinds of proofs. The one presented was constructed by T. Mankus, and the one not presented is constructed by G. K. Manacher. Manacher's proofs involve the formation of two sets as by-products of the execution of the CS/HP and CS/HC algorithms. These sets can be considered a pseudo-bipartite graph. Either this pseudo-bipartite graph has a HP or HC, or in the event that the CS algorithms fail, can trivially be shown not to have respectively a HP or HC. This line, perhaps for the first time, achieves a nonconstructive proof for the existence of a canonical example given the existence of any example.

Appendix A.

CS/HC ALGORITHM FOR HC PRODUCTION

```
Input: Ordered listing of the intervals in I;
Output: Two ordered listings of intervals that join to form a HC, or
        Failure;

1)  Set $S_{1,0} \leftarrow \{1,2,...,n-1\}$ ; Set $U_0 \leftarrow \infty$ ; Set $L_0 \leftarrow \infty$ ; Set $U_1 \leftarrow n$;
    Set $k \leftarrow 1$; Set $j \leftarrow 0$;
2)  FOR i=2 TO n DO;
3)     Set w to equal the largest index in $S_{k,j}$ that intersects $U_k$ or $L_j$.  If no
       such w exists, then Report Failure; and STOP;  If one exists, call it w.
4)     If w intersects $U_k$ but not $L_j$ , call it $U_{k+1}$ .
5)     If w intersects $L_j$ but not $U_k$ , call it $L_{j+1}$ .
6)     If w intersects both $U_k$ and $L_j$ , set it to $U_{k+1}$ if $U_k$ is larger or $L_{j+1}$ if
       $L_j$ is larger.
7)     Remove w from $S_{k,j}$ .
8)     Reset k and j as necessary.
    END DO;
9)  IF($U_k$ does not intersect $L_j$) THEN  Report Failure;
    ELSE  $U_1$__$U_2$__ ... $U_k$__$L_j$__$L_{j-1}$ ... $L_1$  is the HC.
```

This algorithm was first discovered by Carol Smith in 1984[S 84]. Smith was then a graduate student at UIC and is now at Bell Laboratories in Naperville, Illinois. In fact, attempts to prove her algorithm correct were the motivation for this paper, and we wish to express gratitude here to her for this inspiration.

REFERENCES

[AHU 74] Aho,Hopcroft,Ullman. "The Design and Analysis of Computer Algorithms". Addison-Wesley Pub. Co., 1974, 87-90,148-152.

[B 83] Bertossi, Alan. "Finding Hamiltonian Circuits in Proper Interval Graphs". Inf. Process. Lett. 17, 1983, 97-101.

[BL 86] Booth, K. S., and Leuker, G. S. "Testing for the Consecutive Ones Property, Interval graphs and graph Planarity using PQ-tree Algorithms". J. Comput. System Sci. 13, 1976, 335-379

[FW 78] Fredman, M. L., and Weide, B. "On the complexity of computing the Measure of a ,b ". Comm. ACM 21, 1978, 540-544.

[K 85] Keil, J. Mark. "Finding Hamiltonian Circuits in Interval Graphs". Inf. Process. Lett. 20, 1985, 201-206.

[S 84] Smith, Carol. "A new algorithm for optimum computation of Hamiltonian Circuits in Interval Models". Private communication, 1984.

[vEB 77] van Emde Boas, P. "Preserving Order in a Forest in less than Logarithmic Time and Linear Space". Inf. Process. Lett. 6, 1977, 80-82.

COMPUTING A CONDITIONAL EDGE-CONNECTIVITY OF A GRAPH

ABDOL-HOSSEIN ESFAHANIAN
Computer Science Department
Michigan State University
East Lansing, MI 48824

S. LOUIS HAKIMI
Department of Electrical and
Computer Engineering/Computer Science
University of California
Davis, California 95616

1. INTRODUCTION

The *conditional edge-connectivity* $\lambda(G:P)$ of graph G has been defined by Harary [3] as the minimum cardinality $|S|$ of a set S of edges such that $G-S$ is disconnected and every component of $G-S$ has the given graph property P [3]. In this paper, we consider $\lambda(G:P)$ for a certain property P. In particular, we obtain lower and upper bounds for $\lambda(G:P)$ for a general graph G when P is defined as follows: A graph H satisfies property P if it contains more than one vertex. Note that if H is a connected graph, condition P implies that H should contain at least one edge. For this reason, the condition P defined as above, will be denoted by $\lambda(G:\{\chi\geq 2\})$ where χ is the chromatic number [1]. We then give an algorithm which computes $\lambda(G:\{\chi\geq 2\})$ for a given graph G. Finally, we present a new generalization of the notion of connectivity.

2. COMPUTING LOWER AND UPPER BOUNDS FOR $\lambda(G:\{\chi\geq 2\})$

For a graph $G(V,E)$, let $I(v)$, $v\in V$, be the set of all edges incident to vertex v in G and, $I(v)$ is called an *incident-set* in G. Similarly, let $A(v)$ be the set of all vertices adjacent to v in G; $A(v)$ is called an *adjacent-set* in G. Then, the *degree* of a vertex v and, the *minimum* and the *maximum degrees* of G are respectively $d(v)=|I(v)|$, $\delta(G)=\min\{d(v)\,|\,v\in V\}$, and $\Delta(G)=\max\{d(v)\,|\,v\in V\}$. The *degree* of an edge $e=(u,v)\in E$, is defined as $d(e)=d(u)+d(v)-2$. Let $\xi(G)=\min\{d(e)\,|\,e\in E\}$. Note that for any graph G we have $\xi(G)\leq\Delta(G)+\delta(G)-2$. For a set $X\subset V$ (or $X\subseteq E$), the notation $<X>$ represents the *induced subgraph* [1] of G with the vertex (edge) set X. An *edge cut* of G is a set S of edges such that $G-S$ is disconnected and no proper subset of S is an edge cut. For a set $W\subset V$, $(W,V-W)$ denotes the edge cut consisting of all edges with one end vertex [1] in W and one end vertex in $V-W$.

It is well-known that for any graph G we have $\lambda(G)\leq\delta(G)$ [1]. It is clear that $\lambda(G)\leq\lambda(G:\{\chi\geq 2\})$ since for the case of $\lambda(G)$ no condition is imposed on the components. We now have the following lemma.

Lemma: If G is a connected graph with at least four vertices and it is not a star graph, then $\lambda(G)\leq\lambda(G:\{\chi\geq 2\})\leq\xi(G)$.

Proof: The left hand inequality is easy to see and therefore we only consider the right hand inequality. Let $e=(u,v)$ be an edge in G such that $d(e)=\xi(G)$. Now define the edge cut $S=(\{u,v\},V-\{u,v\})$. Note that $|S|=\xi(G)$ and if $G-S$ has no trivial (i.e. K_1) component we are done. Thus, we consider the following two cases. Let $H=G-\{u,v\}$.

Case 1: The subgraph H has at least one component which is not K_1. Let X be the vertex set of this component. Now define a new edge cut for G, namely, $S'=(V-X,X)$. It can easily be seen that $G-S'$ is disconnected and has no component which is K_1. As $|S'|<|S|=\xi(G)$, the lemma is proved for this case.

Case 2: Every component of H is K_1. Define $N_u=\{w\in V(H)|$ in G, w is adjacent to u but not to $v\}$, $N_v=\{w\in V(H)|$ in G, w is adjacent to v but not to $u\}$, and $N_{uv}=\{w\in V(H)|$ in G, w is adjacent to both u and $v\}$. We first show that N_{uv} must be empty. Assume otherwise. This condition and $|V|\geq 4$ imply that in graph G, we either have $d(u)\geq 3$ or $d(v)\geq 3$ or both. Without loss of generality, assume $d(v)\geq 3$ and, for a vertex $x\in N_{uv}$, let $e'=(x,u)\in E(G)$. Since in G we have $d(x)=2$, we must then have in G

$$d(e')=d(u)+d(x)-2=d(u)<d(u)+d(v)-2=d(e)=\xi(G),$$

which is contrary to our assumption. Now, having $|N_{uv}|=0$, if either of the two sets N_u and N_v is an empty set, G is a star graph, again contrary to our assumption. Therefore, the only remaining situation to consider is when $|N_{uv}|=0$, $|N_u|\geq 1$, and $|N_v|\geq 1$. Given this situation, it is clear that the edge $e=(u,v)$ is a *bridge* [1] in G such that $G-\{(u,v)\}$ is disconnected and has no K_1 component. As in this situation $\lambda(G:\{\chi\geq 2\})=1\leq\xi(G)$, the lemma holds for this case also. □

It should be stated that the above upper bound for $\lambda(G:\{\chi\geq 2\})$ is tight in the sense that there are infinitely many graphs for which the equality holds. An example is the class of *n-cube* graphs [1,6].

3. COMPUTING $\lambda(G: \{\chi \geq 2\})$ FOR A GIVEN GRAPH G

In this section, we develop an algorithm which computes $\lambda(G: \{\chi \geq 2\})$ for a given graph G which satisfies the statement of Lemma 1.

All the known algorithms for the computation of the classical connectivities of a graph are based on solving a number of network flow problems [4]. We will use the same technique in computing $\lambda(G: \{\chi \geq 2\})$.

Hereafter assume that $G(V, E)$ is a connected graph for which $\lambda(G: \{\chi \geq 2\})$ exists. Let S, $S \subset E$, be the desired edge cut, i.e., $|S| = \lambda(G: \{\chi \geq 2\})$ and $G - S$ is disconnected and has no K_1 components. It is clear that $G - S$ has exactly two components and let L and R be the vertex sets of these components. We know, of course, that $|L| > 1$ and $|R| > 1$ and, $< L >$ and $< R >$ are connected graphs and therefore both graphs contain at least one edge of G. This suggests an alternative way of defining $\lambda(G: \{\chi \geq 2\})$ as described below.

Two edges in G are said to be *independent* if they do not have a common end vertex. Let e_i and e_j be independent edges in G. An *$(e_i - e_j)$ edge separator* is a set of edges $S \subseteq E - \{e_i, e_j\}$, such that the removal of S from G will destroy every path from any end vertex of e_i to any end vertex of e_j. Let $M(e_i - e_j)$ represent the least cardinality of an $(e_i - e_j)$ edge separator. It is not difficult to see that

$$\lambda(G: \{\chi \geq 2\}) = \min\{M(e_i - e_j) \mid \text{unordered independent pairs } e_i, e_j \in E\}.$$

We proceed with the computation of $M(e_i - e_j)$. Let $e_i = (u, v)$ and $e_j = (x, y)$. Note that the end vertices u, v, x, and y are all distinct as we are assuming that e_i and e_j are independent edges in G. Now, contract the edge e_i in G by identifying its end vertices u and v to obtain a single vertex called s. Do the same thing for edge e_j and call the corresponding vertex t. The resulting graph will be referred to as G'. We now define a network [2] N on graph G' by letting vertex s be its source and vertex t be its sink. Then, each edge is replaced by two oppositely directed arcs [2], the capacity of each arc is set to unity. It is not difficult to see that the value of a max-flow function [2] in network N is equal to $M(e_i - e_j)$ in graph G. Given this, $M(e_i - e_j)$ can be computed by solving a network flow problem in N. This and other observations led us to the following algorithm for the computation of $\lambda(G: \{\chi \geq 2\})$. More details can be found in the full paper.

Algorithm for the computation of $\lambda(G: \{\chi \geq 2\})$ for a given graph $G(V, E)$

```
0. If $|V| \leq 3$ or $G$ is a star graph, STOP, $\lambda(G: \{\chi \geq 2\})$ does not exist.
1. Select an edge $e \in E$ and let $e = (u, v)$.
2. Compute $\lambda_1 = \min\{M(e - e_i) | e_i \in E - e$ and , $e$ and $e_i$ are independent in $G\}$.
3. Compute $\lambda_2 = \min\{M(e_i - e_j) | e_i \in I(u) - e, e_j \in I(v) - e$ and, $e_i$ and $e_j$ are independent in $G\}$.
4. $\lambda(G: \{\chi \geq 2\}) \leftarrow \min(\lambda_1, \lambda_2)$, STOP.
```

4. ANOTHER CONNECTIVITY GENERALIZATION

Harary formulated the notion of conditional connectivity by imposing conditions on the components of $G - S$. We introduce the notion of *restricted connectivity* by imposing conditions on the set S. Formally, the *R-edge-connectivity* $\lambda(G|S: R)$ of a graph $G(V, E)$ is the minimum cardinality $|S|$ of a set S of edges such that $G - S$ is disconnected and S is restricted to a given set R of subsets of E. The *R-vertex-connectivity* $\kappa(G|S: R)$ can be defined similarly with S being a set of vertices and R being a set of subsets of V. Note that when $R = \{X \subset E | \text{for any } v \in V, I(v) \not\subset X\}$, then we have $\lambda(G: \{\chi \geq 2\}) = \lambda(G|S: R)$. Some observations regarding $\lambda(G|S: R)$ and $\kappa(G|S: R)$ are given in the full paper. For a thorough treatment of the notion of connectivity, its many generalizations and related problems, see reference [5].

5. REFERENCES

[1] F. Harary, *Graph Theory*. Addison-Wesley, Reading, MA, 1972.

[2] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*. North Holland, New York, N.Y., 1980.

[3] F. Harary, "Conditional Connectivity," *NETWORKS*, Vol. 13 (1983), pp. 346- 357.

[4] A-H. Esfahanian and S.L. Hakimi, "On Computing the Connectivities of Graphs and Digraphs," *NETWORKS*, Vol. 14 (1984), pp. 355-366.

[5] O.R. Oellermann, "Generalized Connectivity in Graphs," *Ph.D. dissertation*, Department of Mathematics, Western Michigan University, Kalamazoo, Michigan, August 1986.

[6] A-H. Esfahanian, "Generalized measures of Fault-Tolerance with Application to n-Cube Networks," submitted to *IEEE Transac. on Comput.*, May 1987.

SMALL-LARGE DYNAMIC PROGRAMMING FOR TREES

James Mc Hugh
Yehoshua Perl

C.I.S. Department
N.J.I.T.
Newark, New Jersey 07102

ABSTRACT

We present a new dynamic programming technique for trees which is a variation on the Left-Right dynamic programming method of Johnson and Niemi. We first introduce the notion of Contact Vertices, as illustrated in the problem of Economical Weighted Domination on a Tree. We show how the complexity of of the Left-Right method applied to this problem depends critically on the number of contact vertices. We then introduce the Small-Large method. We establish a logarithmic upper bound on the contact vertices, and present a Labelling algorithm that enables us to minimize the maximum number of contact vertices that arise. We then apply the method to obtain an improved solution to the Economical Weighted Domination problem.

1. INTRODUCTION.

We present a new dynamic programming technique for trees. The technique is a variation on the Left-Right Dynamic Programming method recently introduced by Johnson and Niemi [3]. The Left-Right method has been successfully used to develop dynamic programming algorithms that have improved performance with respect to a problem parameter which is non-polynomial in the size of the problem, in exchange for potentially inferior performance with respect to a problem parameter which is polynomial in the size of the problem. For example, [6] uses the Left-Right method to obtain an algorithm of performance $O(cn^4)$ rather than the $O((c^2)(n^3))$ performance obtained using the usual bottom-up dynamic programming technique, where n is the number of vertices in a graph and c is a bound on the edge capacities. In the case that $c > O(n)$, the Left-Right performance is asymptotically superior.

The technique that we will describe was suggested by an attempt to adapt the Left-Right method to a generalized vertex domination problem on trees that we considered in [4,5]. That problem, the problem of Economical Weighted Domination, was defined as follows. Let G(V,E) be a graph. A vertex v in V(G) is said to dominate both v itself and the vertices adjacent to v. A set of vertices S in G dominates

---

Supported in part by the State of New Jersey SBR Grants.

all the vertices in the union of the vertex sets dominated by the member vertices of S. We denote by D(S) the set of vertices dominated by S. We will suppose that associated with each vertex v in G there are a pair of parameters: The cost c(v) of the vertex and the population or weight w(v) of the vertex. The cost c(S) of a set of vertices S is defined as the sum of the costs of the member vertices of S. The population w(S) dominated by S is defined as the sum of the populations of the vertices in D(S). The population of a vertex in D(S) which is dominated by more than one vertex in S is counted only once. We assume there is a Budget constraint B that limits the amount that can be expended on vertex costs. The problem of Economical Weighted Domination is then defined as the problem of finding a set of vertices S in V(G) that maximizes the dominated population w(S) subject to the constraint that c(S) <= B, the budget constraint. An application for this problem appears in optimizing the locations of service centers in a network of cities. We consider the case that G is a tree. The problem is shown in [4,5] to be NP-Hard, where a pseudo-polynomial dynamic programming procedure is presented. We refer to [2] for further references on domination, and to [1] for domination on trees in particular.

A careful analysis of the dynamic programming methodologies we used in [5] suggested the notion of Contact Vertices. These are vertices at which dynamic programming subproblems, which have to be combined using the principle of optimality, interface with one another. For example in the propagation method of dynamic programming, we extend partial solutions from a smaller to a larger subtree by appending a new subtree at the root v of the original subtree. In this case, the roots of the subtrees are contact vertices. In the Left-Right method of dynamic programming, we extend partial solutions from a subtree to a larger subtree by appending a single new vertex to a previously solved subtree. In this case, ALL of the vertices where the subtree can be extended are contact vertices.

In both of these methods, the extended solutions arise essentially by combining existing optimal solutions for subtrees at contact vertices. However, a contact vertex can be in any one of several states: dominating, dominated but not dominating, or neither dominating nor dominated. It is critical to observe that we have to know the states the contact vertices are in to combine the subtree solutions correctly. Specifically, we must know the domination state of each current contact vertex of the subtree solutions to correctly calculate the population dominated by their combination.

Because of this state dependence, if a subtree has H contact vertices, there will be 3^H solutions to be maintained, one per possible contact vertex state. The exponential dependence on H implies we should try to minimize H. Since the depth first order according to which a tree is scanned by the Left-Right method is arbitrary, we can try to find an order which minimizes H, while still applying the basic idea of the Left-Right method. For the relatively straightforward Left-Right adaption used in [5], we observed that in the special case of complete balanced binary trees using the Left-Right depth first traversal, the maximum value of H encountered during the sequence of subproblems was logarithmic in the size of the tree, and worthwhile performance improvements could be obtained. However, for an arbitrary tree, straightforward adaption of the Left-Right traversal would lead to unsatisfactorily large values for H.

Certainly, the number of contact vertices is highly dependent on the traversal order. For example, consider the tree T constructed as follows. Let $v_1 \ldots v_k$ be a linear list of vertices, where v_1 is the root of the tree. Assume each vertex v_i, i>1, is the left child of v_{i-1}. Assume also that each vertex v_i, i>1, has one right child w_i. If we traverse T so that $v_1 .. v_k$ are traversed in that order, as we would by the Left-Right method, then the number of active contact vertices by the time we reach v_k is k. Consequently, this will entail keeping solutions for 3^k contact vertex states. On the other hand, if we traverse T in the order v_1, w_1, v_2, $w_2, \ldots, v_{k-1}, w_{k-1}$, there will never be more than one contact vertex at any point, so H will be at most one, and no more than 3 state dependent solutions will have to be maintained, a dramatic improvement.

In the present paper, we show that for any tree, for which we require an optimal solution, it is possible to build the tree, one vertex at a time, as required by the basic Left-Right strategy, in such a way as to keep the number of contact vertices small. We begin with the case of binary trees, and extend our results to n-ary trees. For binary trees, we establish an O(lg(N)) bound on the size of H, where N is the number of vertices in the tree. We then obtain a similar bound for the general n-ary case. We also show how to find the optimal value of H using a labelling algorithm, and then show how to use the resulting labelling to process the tree, smaller subtrees first, to obtain a traversal optimizing H. The manner of operation of the algorithm suggests the name Small-Large Dynamic programming for our method. Interestingly, the labelling algorithm used to calculate H for a rooted tree is the same as the labelling algorithm used to calculate the minimum number of registers needed to evaluate an expression without stores to memory, a classical result of Sethi and Ullman [7]. However, both the initialization for the labelling and our underlying problem are quite different. We also present a graph-theoretic characterization for the minimax value of H. Finally, we apply these results to obtain an improved solution to the Economical Weighted Domination problem. Although we illustrate the technique with respect to the Weighted Domination problem, the approach shown here appears to have application to other problems.

2. UPPER BOUND FOR MINIMAX NUMBER OF CONTACT VERTICES.

We will establish an upper bound on the number of contact vertices that arise for an appropriate edge by edge construction of a tree. For convenience, we restrict ourselves to binary trees, though the same bound applies as well to N-ary trees. We will show a binary tree can always be built in an edge by edge fashion in such a way that the maximum number of contact vertices that arise during the construction of the tree is bounded by lg (n), where n is the size of the tree. The following terminology will be useful. We define a Construction Sequence $\{T_i: i=1,..,n\}$ for a rooted binary tree T of order n rooted at a vertex v as a sequence of subtrees T_i of T, such that: each subtree T_i has order i; T_i is contained in T_{i+1}, for $i < n$; $T_1 = \{v\}$, and $T_n = T$. Thus, each subtree T_i is obtained from its predecessor subtree T_{i-1} by adding to some vertex u in T_{i-1} an edge (u,v) of T not lying in T_{i-1}. We define a vertex w in T_i to be a Contact Vertex for T_i if there is an edge (w,z) in T, for which z is in $T-T_i$. We denote the number of contact vertices of T_i by H_i. We will denote the maximum value of H_i for a given Construction sequence $\{T_i,i=1..n\}$ by $H(\{T_i,i=1..n\})$. We will denote the minimum value of $H(\{T_i,i=1..n\})$ taken over all possible construction sequences for T by H(T). We let lg(n) denote the greatest integer in the ordinary base 2 real valued log(n). Using this terminology, we may state our bound as follows:

Theorem (Minimax Upper Bound) Let T be a binary tree of order n. Let H(T) denote the minimax number of contact vertices over all construction sequences for T. Then, $H(T) <= lg (n)$.

Proof: The proof is by induction. We assume the Theorem is true for m, $m < n$, and prove it for $m = n$. The idea of the proof is as follows. If the root has only one child v_1, then $T_1=\{v\}$, $T_2=\{v,v_1\}$, and now we process the subtree rooted at v_1 instead of v, and v is no longer a contact vertex. We first observe that one of the subtrees rooted at a child of the root v of T has order at most n/2. Therefore, we may assume by induction that we can build this subtree, starting at its root, and using at most lg(n/2) contact vertices. If we include the root v of T as well, then at most lg(n/2)+1 or lg(n) contact vertices occur. At the point at which we have constructed the subtree consisting of the selected subtree of v together with v, there remains only a single contact vertex, v. We can then complete the construction of T by extending the construction into the remaining subtree. The first step, which adds the remaining edge at v and its other endpoint to the subtree constructed so far, removes v as a contact vertex. The remaining construction, once again by induction, never introduces more than lg(n) contact vertices. This completes the proof of the bound. A similar proof applies in the case of N-ary trees.

Figure 1 illustrates the Minimax Upper Bound Theorem for a Fibonnaci tree. The tree is of size 12, so the a priori upper bound on H is lg(12) or 3. If we process the tree using the procedure in the proof, then the maximum number of contact vertices we obtain is 2. On the other hand, for a complete binary tree of k levels, H(T) equals k-1, where the tree size n equals $2^k - 1$.

It is important to note that the minimax number of contact vertices for a construction for T depends on what vertex is taken as the root of T. Therefore, to find the overall optimal value, we must consider constructions with each possible root. On the other hand, the logarithmic bound holds regardless of what vertex is chosen as the root. It is also interesting to observe that the minimax number of contact vertices in any construction sequence is attained by a subtree construction sequence, of the type we have described, as opposed to a less restrictively defined construction sequence whose sequence members were allowed to be forests. In particular, we can always transform a forest construction to a subtree construction of the kind we have defined without increasing the maximum number of contact vertices that arise for the sequence. In any case, for the application that we have in mind, the construction sequence must be a subtree sequence of the type we have defined in order to satisfy an underlying requirement of the Left-Right method for which partial solutions to the problem are extended only by adding a single vertex at a time. This constraint is violated by a forest construction since at some time during such a construction we would have to combine two nontrivial trees, rather than a tree and a single vertex, defeating the purpose of the Left-Right technique.

3. LABELLING ALGORITHM FOR FINDING THE MINIMUM VALUE H(T).

The upper bound technique of section 2 suggests a labelling algorithm which computes the optimal value of H(T) for a given rooted tree T and which also determines a corresponding optimal construction sequence for T. Let T be rooted at v as before, and let $T(x_1)$ and $T(x_2)$ denote the subtrees of T rooted at the children x_1 and x_2 of v. Then, H(T) satisfies the following relation:

$$H(T) = \begin{cases} \max\{H(T(x_1),H(T(x_2))\} & \text{if } H(T(x_1)) <> H(T(x_2)) \\ \max\{H(T(x_1),H(T(x_2))\}+1 & \text{if } H(T(x_1)) = H(T(x_2)) \end{cases} \qquad (1)$$

We will initialize the leafs of the tree to 0. If a leaf is an only child of its parent, then we will also initialize the parent of the leaf to 1. If we then apply this relation to T in a bottom-up fashion, then each vertex in T is labelled with the value of H for the subtree rooted at that vertex.

The correctness of this labelling algorithm for H(T) follows from an application of the principle of optimality. Our objective is to find the maximum number of contact vertices in an optimal construction sequence for a tree rooted at v. Certainly every subtree in any construction sequence must contain the root v. Furthermore, we may assume without loss of generality that an optimal construction completely fills out one of the subtrees rooted at one of the children of v, before it expands into the other subtree rooted at the remaining child of v. Because, any construction not of this form can be converted to one in this form without increasing its maximum number of contact vertices, merely by adding the edges of one of the children's subtrees in the same order they as are added in the original construction, and then adding the edges for the remaining subtree in a similar order. Suppose that $T(x_1)$ is the first subtree filled out by the resulting construction, and that $T_1 \ldots T_m$ are the subtrees for this leading part of the construction. That is, T_m is $T(x_1) \cup \{v\} \cup \{v,x_1\}$. The sequence formed by intersecting $T(x_1)$ with each of the subtrees T_i, $i=1..m$, determines a construction sequence for $T(x_1)$. We can assume without loss of generality that this construction is optimal for $T(x_1)$. A similar assumption can be made regarding the remaining subtree $T(x_2)$. Thus, an optimal construction, when restricted to either of the subtrees $T(x_i)$, $i=1,2$, is locally optimal. It only remains to determine in which order the subtrees should be filled. But, this is precisely the decision which is handled by the recurrence relation. Referring to Figure 2, we add the subtree with the smaller value of H first, since this results in an overall smaller maximum number of contact vertices for the construction. The validity of the recurrence relation then follows.

Similar reasoning readily establishes a related result for an N-ary tree T. Let T be rooted at v as before, and let $T(x_i)$ denote the subtrees of T rooted at the children x_i, $i=1..N$ of v. Then, H(T) satisfies:

$$H(T) = \begin{cases} \max\{H(T(x_i)), i=1..n\} & \text{if there is a unique maximum entry } H(T(x_m)), \\ \max\{H(T(x_i)), i=1..n\}+1 & \text{if there is more than one entry of maximum magnitude.} \end{cases} \qquad (2)$$

Optimal Construction Algorithm

We can use the Labelling Algorithm to compute an optimal construction sequence. The procedure is as follows:

(1) Label the vertices of the tree using the labelling algorithm.

(2) Once the root of the tree is labelled, define an optimal construction by adding vertices in the following order:

   (a) The root vertex v.
   (b) Recursively add the vertices lying in the subtree of the smaller labelled child of v.
   (c) Recursively add the vertices lying in the subtree of the larger labelled child of v.

Ties are broken arbitrarily. The vertices of the Fibonnaci tree in Figure 1a are shown with their H values in parentheses, and all vertices are numbered according to the order in which they are added in an optimal construction sequence.

4. APPLICATION TO ECONOMICAL WEIGHTED DOMINATION.

We will now show how optimal construction sequences can be used to obtain an efficient dynamic programming algorithm for the economical weighted domination problem. Let $\{T_i\}$ be an optimal construction sequence for T, and suppose the vertices $\{v_i\}$ of T are indexed according to the order in which they are added in the construction sequence. Each subtree T_i has an associated set of contact vertices C_i. Each vertex in C_i can be in any of three states:

State-1: Dominating,
State-2: Dominated but not dominating, or
State-3: Neither dominating nor dominated.

Let Status(C_i) be a vector of length equal to the cardinality of C_i and whose j^{th} entry define an instance of a state requirement on the j^{th} vertex of C_i. Let W(T_i,Status(C_i),B) denote the weight of a maximum weight dominating set in T_i of cost at most B where the vertices of C_i are in the states given by an instance of the status vector Status(C_i). If a particular combination of states is unrealizable, then the corresponding problem is considered to be undefined. For example, in Figure 1, if v_1 is in state 1 (dominating), then v_2 cannot be in state 3 (not dominated). We will use the notation x(k) to indicate a contact vertex x is in state k. We will use the notation Status(C_i+x(k)) to indicate that a contact vertex x in state k already lies in or is to be added to the state vector Status(C_i).

The dynamic programming relations for W are as follows. Let (u,v) be the edge to be appended to T_i, necessarily at a contact vertex v of T_i, in order to obtain T_{i+1}. We must determine the constrained maximum weight solution for every status vector instance in this extended tree.

If u and v are both dominating, then the recurrence relation is just:

$$W(T_iU\{u\},Status(C_i+v(1)+u(1)),B) = W(T_i,Status(C_i+v(1)),B-c(u))+w(u). \qquad (3.1)$$

If u is dominating, but v is not, then v must be dominated in the extended solution. There are two ways in which this can happen, as indicated by the recurrence:

$$W(T_iU\{u\},Status(C_i+v(2)+u(1)),B) = \max \{W(T_i,Status(C_i+v(2)),B-c(u))+w(u), \quad W(T_i,Status(C_i+v(3)),B-c(u))+w(v)+w(u)\}. \qquad (3.2)$$

If u is not dominating but v is, then u must be dominated in the extended solution and the recurrence relation is:

$$W(T_iU\{u\},Status(C_i+v(1)+u(2)),B) = W(T_i,Status(C_i+v(1)),B) + w(u). \qquad (3.3)$$

If both u and v are not dominating, but v is dominated, then:

$$W(T_iU\{u\},Status(C_i+v(2)+u(3)),B) = W(T_i,Status(C_i+v(2)),B). \qquad (3.4)$$

If neither u nor v are dominating or dominated, then:

$$W(T_iU\{u\},Status(C_i+v(3)+u(3)),B) = W(T_i,Status(C_i+v(3)),B). \qquad (3.5)$$

This shows how to extend a solution from T_i to T_{i+1}. In order to maintain the desired bounds on the number of contact vertices, we also have to collapse solutions when a vertex ceases to be a contact vertex. The procedure for this is simple. It is easy to show that the overall algorithm has complexity $O(B(n^3)3^{H(T)})$, which is linear in B and polynomial in n, in contrast with the usual $O(nB^2)$ method. Note that the complexity is $O(Bn^2)3^{lg(n)})$ if we use a suboptimal value for the maximum number of contact vertices in the tree construction, as demonstrated by the Minimax Upper Bound Theorem.

5. CONCLUDING REMARKS.

In this paper, we have introduced a new tree processing technique, called Small-Large Dynamic Programming, which is based on the notion of Contact Vertices. The performance of the technique depends on the number of contact vertices that arise during the construction of a problem tree. We have given an upper bound on the minimax number of contact vertices that can occur, provided the tree is constructed properly, and a Labelling algorithm for the optimal construction of the tree. Finally, we have shown how Small-Large Dynamic Programming allows a solution to the Weighted Domination problem on a Tree which is linear in the budget constraint. This approach appears to have application to other similar problems.

REFERENCES

[1] Hedetniemi, S.,Hedetniemi,S. and Laskar,R., Domination in Trees: Models and Algorithms, in Graph Theory and its Applications to Algorithms and Comp.Sci.,ed.Y.Alavi,et al.,John Wiley,1985,423-441.

[2] Hedetniemi, S., and Laskar, R., A Bibliography on Domination,Clemson Univ., Nov. 1985.

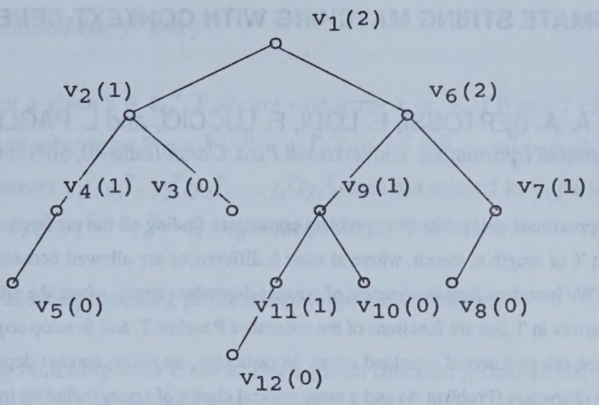
[3] Johnson, D.S., and Niemi, K.A., On Knapsacks,Partitions, and a New Dynamic Programming Technique for Trees, Math. Operations Res.,8(1983), 1-14.

[4] McHugh,J.,and Perl,Y.,Maximum Economical Domination in Tree Networks,Proc. 24th Ann.Allerton Conf.on Commun. Cont. and Comp.,1986, 69-78.

[5] Mc Hugh, J., and Perl, Y., Best Location of Service Centers in a Treelike Network under Budget Constraints, to appear in Annals of Discrete Mathematics.

[6] Perl,Y.,and Snir,M.,Circuit Partitioning with Size and Connection Constraints, Networks, 13(1983),pp.512-516.

[7] Sethi,R., and J.D. Ullman, "The Generation of Optimal Code for Arithmetic Expressions." J.ACM, 17, 4(1970), 715-728.


$v_1(2)$
$v_2(1)$
$v_6(2)$
$v_4(1)$
$v_3(0)$
$v_9(1)$
$v_7(1)$
$v_5(0)$
$v_{11}(1)$
$v_{10}(0)$
$v_8(0)$
$v_{12}(0)$

Notation: Vertex-Identifier (H Label)

(a) Fibonnaci Tree Rooted at v_1.

#C.V.	C.V.	Subtree
1	v_1	v_1
2	v_1, v_2	v_1, v_2
2	v_1, v_2	v_1-v_3
2	v_1, v_4	v_1-v_4
1	v_1	v_1-v_5
1	v_6	v_1-v_6
2	v_6, v_7	v_1-v_7
1	v_6	v_1-v_8
1	v_9	v_1-v_9
1	v_9	v_1-v_{10}
1	v_{11}	v_1-v_{11}
0	-	v_1-v_{12}

(b) Trace for (a).

Figure 1. Fibonnaci Tree Example.

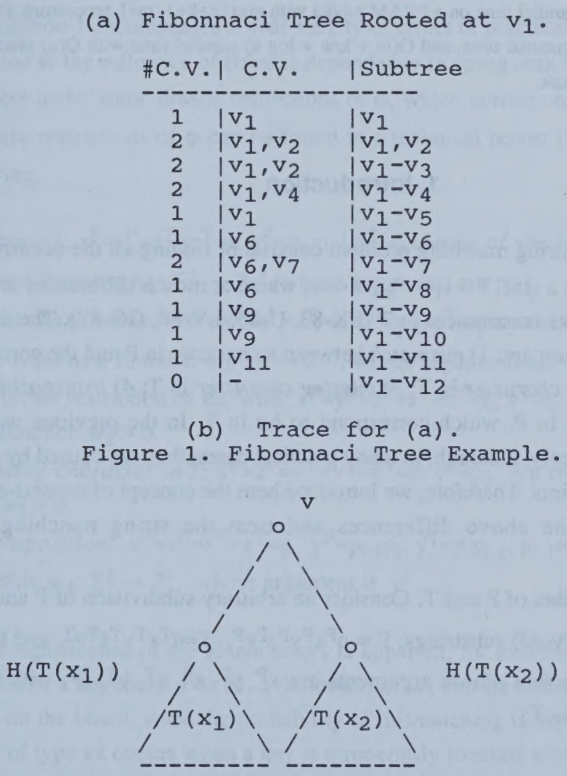

v
$H(T(x_1))$
$H(T(x_2))$
$T(x_1)$
$T(x_2)$

Figure 2. Diagram for Recurrence Relation.

APPROXIMATE STRING MATCHING WITH CONTEXT-DEPENDENCY

A. A. BERTOSSI, E. LODI, F. LUCCIO, and L. PAGLI
Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56100 Pisa, Italy

Abstract. The approximate string matching problem consists of finding all the occurrences of a pattern P of length m in a text T of length n, $m<<n$, where at most k differences are allowed between P and each of its occurrences in T. We introduce here the concept of context-dependent errors, where the differences between P and T depend on errors in T that are functions of the context of P and/or T, and develop sequential and parallel algorithms for some relevant sets of weighted errors. In particular, we allow: context-dependent mismatches, extra and missing characters (Problem A) and a more general choice of errors including transpositions of two consecutive characters (Problem B). A set of theoretical results allows to solve Problem A in O(kn) sequential time, and in O(k + log m) parallel time on a PRAM model with $\max\{n+k+1, m^2\}$ processors. Problem B is instead solved in O(mn) sequential time, and O(m + k/w + log n) parallel time with O(n) processors on a feasible bounded degree network.

1. Introduction

The approximate string matching problem consists of finding all the occurrences of a pattern $P = p_1 \cdots p_m$ in a text $T = t_1 \cdots t_n$, $m<<n$, where at most k differences are allowed between P and each of its occurrences in T [SK-83, U-85, LV-86, GG-87]. The differences considered in the literature are: 1) *mismatch* between a character in P and the corresponding character in T; 2) *extra character* in T; 3) *missing character* in T; 4) *transposition* of two adjacent characters *ab* in P, which correspond to *ba* in T. In the previous works these differences can occur freely, although, in many practical cases, they are caused by errors that obey to specific constraints. Therefore, we introduce here the concept of *context-dependent errors* which cause the above differences, and treat the string matching problem accordingly.

Let Σ be the alphabet of P and T. Consider an arbitrary subdivision of P and T in five concatenated (possibly void) substrings: $P = u^Px^Py^Pz^Pv^P$, $T=u^Tx^Ty^Tz^Tv^T$, and let $\varphi: \Sigma^{*6} \rightarrow \mathbb{Z}_0^{+}$ be a *weight function* whose arguments are x^P, y^P, z^P, x^T, y^T, z^T, with $\varphi(x^P, y^P, z^P, x^T, y^T, z^T)=0$ iff $y^P=y^T$.

Definition 1. y^T is a *context-dependent error of weight w* for y^P (briefly, y^P/y^T is a *w-error*) if $w=\varphi(x^P, y^P, z^P, x^T, y^T, z^T)$.

If $y^P=y^T$, y^P/y^T is a 0-error, also called *match*. Note that a 0-error is independent of the substrings surrounding y^P and y^T.

Definition 2. For a given $k \in \mathbb{Z}_0^+$, P *occurs with error* k *in* T, if P and T can be written as concatenations of substrings: $P = y_1^P y_2^P \cdots y_r^P$, $T = u\, y_1^T y_2^T \cdots y_r^T v$, such that the sum of weights of the errors $y_1^P/y_1^T, y_2^P/y_2^T, \ldots, y_r^P/y_r^T$ does not exceed k. In particular, P occurs *in position* h if $u\, y_1^T y_2^T \cdots y_r^T = t_1 \cdots t_h$ is the prefix of T of length h.

The general string matching problem is then posed as follows:

Problem: *String matching with error* k. For a given function j, find all the positions of the occurrences with error k of a pattern P in a text T.

Definition 1 encompasses a wide variety of errors of practical interest. However, as an introduction to the influence of context-dependency in string matching, the problem will be treated here under some drastic restrictions of φ, which correspond to the following errors (less severe restrictions of φ can be found in a technical report [BLLP-87]). Let ε be the empty string.

e1. *Mismatch*: $x^P=z^P=x^T=z^T=\varepsilon$; $y^P=p_i$ and $y^T=t_j$ consist of single characters. We restrict φ to the new function $w_1: \Sigma^2 \rightarrow \mathbb{Z}_0^+$, whose arguments are p_i, t_j.
e2. *Extra character in T*: $x^P=y^P=z^P=z^T=\varepsilon$; $x^T=t_{j-1}$, $y^T=t_j$ consist of single characters. We restrict φ to the new function $w_2: \Sigma^2 \rightarrow \mathbb{Z}^+$, whose arguments are t_{j-1}, t_j.
e2'. A further restriction of **e2**, with: $x^P=y^P=z^P=x^T=z^T=\varepsilon$, $y^T=t_j$, and φ restricted to the constant function $w_2 > 0$.
e3. *Missing character in T*: $x^P=z^P=x^T=y^T=z^T=\varepsilon$, $y^P=p_i$. We restrict φ to the constant function $w_3 > 0$.
e4. *Transposition*: $x^P=z^P=x^T=z^T=\varepsilon$, $y^P=p_{i-1}p_i$, $y^T=p_ip_{i-1}$, $p_{i-1} \neq p_i$. We restrict φ to the new function $w_4: \Sigma^2 \rightarrow \mathbb{Z}^+$, whose argument is y^P.

The significance of the above errors is apparent, for example, if the text T has been typed through a keyboard. For **e1**, a definition of w_1 can be based on the positions of any two keys on the board, since the probability of mismatching is higher between closer keys. An error of type **e2** occurs when a key is erroneously touched after a neighbor key. Again, the value of w_2 depends on the distance of the two keys. **e2** can also occur if a key is hold down too long, causing a repetition of the same character. Errors **e2'** and **e3** are in fact context independent. They comply with the assumption that any character can be erroneously inserted or missed in the text, without constraints (in this respect, **e3** is more realistic than **e2'**). These two errors correspond to the classical extra and missing characters. Finally, for **e4** the different values of w_4 correspond to transpositions of characters which occur with different probabilities.

In this paper, we restrict our study to the following two instances of the string matching problem.

Problem A. Find all the occurrences with error k of P in T, under the errors **e1**, **e2'** and e3. (This is an extension of the classical problem of [LV-86], to include context-dependent errors of type **e1**)

Problem B. Find all the occurrences with error k of P in T, under the errors **e1**, **e2**, **e3** and **e4**.

We solve Problem A in O(kn) sequential time, and in O(k + log m) parallel time on a PRAM model with max{n+k+1, m^2} processors. Problem B is instead solved in O(mn) sequential time, and O(m + k/w + log n) parallel time with O(n) processors on a feasible bounded degree network.

2. Basic definitions

The classical algorithms known in the literature dealing with approximate string matching are usually based on dynamic programming techniques. In particular it is useful to define an (m+1)x(n+1) matrix D.

Definition 3. For all i,j, $1 \leq i \leq m$, $1 \leq j \leq n$, $D_{i,j}$ is the minimum value of the error between $p_1 \ldots p_i$ and any consecutive substring of the text T ending at t_j.

Matrix D will be bordered with a row 0 and a column 0, corresponding to extremal cases in which either P or T are empty. For any j, $D_{0,j}$ is equal to 0, since an occurrence of an empty pattern can always be recognized at any position of the text. Instead, $D_{i,0}$ has a value related to i, since i characters of the pattern induce i errors of type **e3** in an empty text. We are interested in the last row of D, because if $D_{m,j} \leq k$ then there must be an occurrence with error k of the pattern in the text that ends at tj.

In specific algorithms each entry $D_{i,j}$ is usually derived in a minimization step involving a small set of entries. A minimization path is thus one path followed by the algorithm. Straightforward methods for computing matrix D take O(mn) time. Whenever entries on the same diagonals of D form non decreasing sequences which increase in unit steps, faster algorithms running in O(kn) time may be derived. In fact these algorithms actually compute at most k different values for each diagonal, namely, those corresponding to entries which produce an increment [LV-86]. Since the values along the diagonals remain constant in presence of matchings of subsequences of P and T, the algorithm must include an efficient way of finding the extremes of such subsequences without scanning them. In particular, the algorithm given in [LV-86] actually computes another matrix, L, whose rows correspond to the diagonals of D and whose columns correspond to the error values in such diagonals. This matrix is evaluated in O(kn) time by using a data structure called "suffixes-tree" which allows to compute in O(1) time a matching of maximal length between two substrings [LV-86].

3. Solving Problem A

The algorithm which computes the matrix D can be directly derived as follows:

Algorithm A

```
    for j := 0 to n do $D_{0,j} := 0$;
    for i := 0 to m do $D_{i,0} := iw_3$;
    for i := 1 to m do
        for j := 1 to n do $D_{i,j} := \min\{ D_{i-1,j-1}+w_1(p_i,t_j),\ D_{i,j-1}+w_2,\ D_{i-1,j}+w_3 \}$.
```

It is easy to check that Algorithm A takes time O(mn).

We now show that even in the presence of pattern-dependent errors **e1** and arbitrary weights w_1, w_2, w_3, as considered in Problem A, the values along the diagonals of D remain nondecreasing. This property allows to use the suffixes tree data structure, to achieve an O(kn) time bound. For Problem A we have:

Theorem 1. $D_{i,j} - D_{i-1,j-1} \geq 0$, *for any pair* i, j.

Proof. The proof is long and tedious and can be found in [BLLP-87]. It is based on the following results:

1) $D_{i,j} - D_{i,j-1} \geq -w_3$, for any pair i, j;
2) $D_{i,j} - D_{i-1,j} \geq -w_2$, for any pair i, j. ■

This theorem allows to solve Problem A with proper extensions of Landau and Vishkin's fast sequential algorithm [LV-86], in time O(kn) and space $O(\max(w_2,w_3)\cdot n)$, and parallel algorithm, in the ideal PRAM model, in O(k+logm) steps with $O(n+m^2)$ processors. Details can be found in [BLLP-87].

3. Solving Problem B

Problem B can be solved sequentially in O(mn) time by computing the matrix D with a modified version of Algorithm A, in which the minimization step is modified in order to take into account errors **e2** and **e4**. We delimit the pattern and the text by $p_0=t_0=\$$, and we pose:

Algorithm B

```
for j := 0 to n do $D_{0,j} := 0$;
for i := 0 to m do $D_{i,0} := iw_3$;
for i := 1 to m do
    for j := 1 to n do
        $D_{i,j} := \min\{ D_{i-1,j-1} + w_1(p_i,t_j),\ D_{i,j-1} + w_2(t_{j-1},t_j),\ D_{i-1,j} + w_3,$
                        if $p_{i-1} \neq p_i$ and $p_{i-1} = t_j$ and $p_i = t_{j-1}$
                            then $D_{i-2,j-2} + w_4(p_{i-1}, p_i)$
                            else $\infty$ $\}$.
```

Problem B can be solved with a parallel algorithm based on divide-&-conquer strategies. To see this, we need the following result:

Theorem 2. *$D_{i,j}$ is independent on the values $D_{i,j'}, D_{i-1,j'-1}, \ldots, D_{1,j'-i+1}$ with*

$$j' \leq \begin{cases} j - iw_3/w_{2,m}, & \textit{for } i \leq k/w_3, \\ j - \lfloor k/w_{2,m} \rfloor - 1, & \textit{otherwise}, \end{cases}$$

where $w_{2,m}$ is the minimum value of w_2.

Proof. To prove the theorem, we only need to show that each value $D_{i,j}$ can be generated through a path which does not include any entry lying on diagonal $d = j^*-i$, with $j^* = j - iw_3/w_{2,m}$, if $i \leq k/w_3$, or $j^* = j - \lfloor k/w_{2,m} \rfloor - 1$, otherwise.

By contradiction, let $D_{r,s}$, with $r \leq i$, $s<j$, and $s-r=d$, be an entry on diagonal d which is included in a minimization path yielding the value $D_{i,j}$.

Assume firstly that $i \leq k/w_3$. Obviously, $D_{i,j} \leq iw_3$, since $D_{i,j}$ could be obtainded through a vertical path starting from $D_{0,j}$. On the other hand, the optimal way of deriving $D_{i,j}$ via a path including $D_{r,s}$ contains exactly i-r matchings and j-s-(i-r) horizontal steps. Thus $D_{i,j} \geq (j-s-(i-r))w_{2,m} + D_{r,s}$. Since $D_{r,s} \geq 0$, we have $D_{i,j} \geq (j-s-i+r)w_{2,m}$. Combining the above inequalities, since by assumption $D_{i,j}$ depends on $D_{r,s}$, $(j-s-i+r)w_{2,m} \leq iw_3$ results. Therefore, $(j-s-i+r) \leq iw_3/w_{2,m}$ and $j-iw_3/w_{2,m}-i < s-r = d$, a contradiction.

Assume now that $i > k/w_3$. In this case, since $iw_3>k$, an argument similar to the one used above yields $(j-s-i+r)w_{2,m} \leq k$. Therefore, $j-k/w_{2,m}-i < s-r = d$ and thus $j - \lfloor k/w_{2,m} \rfloor - 1 - i < d$, a contradiction. ■

As an example, consider the matrix D shown in Fig. 1, where m=7, k=8, $w_3=2$ and $w_{2,m}=1$. Then the entries on diagonal d (shown by black squares), as well as those on its left, do not influence those in the gray zone. Thus one could apply the sequential dynamic programming algorithm to the submatrix D' (bordered as usual with an uppermost row of zeroes and a leftmost column whose i-th entry is equal to iw_3) and be guaranteed that the entries of D in the gray zone hold the correct values.

This allows to apply a *divide-&-conquer* strategy to the computation of D, which in turn leads to the design of an efficient parallel algorithm as follows.

We consider n+m−1 processors and assign one processor per diagonal of D. The processors are indexed $P_{-m+1}, P_{-m+2}, \dots, P_{-1}, P_0, P_1, \dots, P_{n-1}$. Processor $P_{d-1}$is devoted to sequentially compute $D_{1,d}, D_{2,d+1}, \dots, D_{m,d+m-1}$, and can communicate only with P_{d-2} and P_d . Each entry $D_{i,j}$, $j-i=d$, is computed by P_d as a function of $D_{i,j-1}$ (which is received from P_{d-1} along with p_i), $D_{i-1,j}$ (which is received from P_{d+1} along with t_j), $D_{i-1,j-1}$ and $D_{i-2,j-2}$ (which have been previously computed by P_d itself). P_d can keep these four values, along with p_i and t_j, in six cells of its memory.

To compute D, we ideally subdivide it into $N = n/(m+\lfloor k/w_{2,m} \rfloor)$ distinct consecutive submatrices $D^1, \dots, D^N$, each having $M = m + \lfloor k/w_{2,m} \rfloor$ columns, as shown in Fig. 2. Informally, the algorithm acts in 3 phases.

Phase 1. The entries of all submatrices D^1, D^3, ... (those having odd indices) are computed in parallel. To do this, each such a D^j is bordered with the usual initial values, namely an uppermost row of zeroes and a leftmost column whose i-th entry is equal to iw_3, by loading such values into the proper processors. By Theorem 2, at the end of Phase 1, the gray zones of D^j (j odd) contain the correct values of D (see Fig. 2).

Phase 2. We reapply the same algorithm of the previous phase to all matrices D^2, D^4, ... (those having even indices). This time, however, the *right* initial values for D^j (j even) are already available in the two rightmost columns of D^{j-1}. Indeed, these values have been computed during Phase 1 and belong to the gray zone of D^{j-1} (see Fig. 2). At the end of Phase 2, each D^j (j even) holds the correct values of D.

Phase 3. We reapply Phase 1 to all the submatrices D^j, j odd. The correct initial values, as computed during Phase 2, are this time available in the two rightmost columns of D^{j-1}. At the end of Phase 3, also each D^j (j odd) holds the correct values of D, and the algorithm terminates.

During each phase, the entries of each D^j are computed in parallel as follows. Let $Q^j_{-m+1}, Q^j_{-m+2}, \dots, Q^j_{-1}, Q^j_0, Q^j_1, \dots, Q^j_{M-1}$be the the M+m-1 processors assigned to the diagonals of D^j. The computation of the entries of a D^j proceeds by anti-diagonals in the following way. At step 1, processor Q^j_0 computes $D^j_{1,1}$; at step 2, processors Q^j_{-1} and Q^j_1 compute, respectively, $D^j_{2,1}$ and $D^j_{1,2}$; at step 3, processors Q^j_{-2}, Q^j_0 and Q^j_2 compute, respectively $D^j_{3,1}$, $D^j_{2,2}$ and $D^j_{1,3}$, and so on. In this way, processor Q^j_x is activated at step $|x|+1$. It starts computing $D^j_{|x|+1,1}$ (if $x<0$) or $D^j_{1,|x|+1}$ (if $x\geq 0$), and then sequentially computes the other entries on its corresponding diagonals at step $|x|+3$, $|x|+5$, etc. The last processor to end is clearly $Q^j_{\lfloor k/w_{2,m} \rfloor}$, which is activated at time $\lfloor k/w_{2,m} \rfloor + 1$, and then takes 2m steps to compute all the entries on its corresponding diagonal of D^j. Since during each phase all the involved D^j's cover entries of D whose diagonals do not overlap, the computation can be carried out in parallel over all the D^j's (j odd or even, depending on the phase), Thus each phase takes $2m + \lfloor k/w_{2,m} \rfloor + 1$ parallel steps. Since there are 3 phases and an intitial O(log N) time is needed to broadcast the i-th character of the pattern p_i to all the Q^j_{i+1} for $1 \leq j \leq N$, the overall parallel time is $O(m + \log N + k/w_{2,m})$.

This parallel algorithm has the good feature of requiring $O(m + \log N + k/w_{2,m})$ time on a feasible bounded degree network of O(n) processors. Indeed, each processor is connected to only two processors and requires only a constant number of memory cells. Moreover, the memory access time for each step of the algorithm is constant, since no memory access conflict among two or more processors can arise.

5. Further research

A variety of open problems remain. Problems A and B are to be seen as examples, albeit important, of the many different situations that may occur in a text-editing or information retrieval environment. Other problems should be considered in the same area, by extending the list of possible errors, and different specifications of context-dependent errors should be given for other fields of application.

References

[BLLP-87] A.A. Bertossi, E. Lodi, F. Luccio and L. Pagli, "String matching with context-dependent errors", Research Report S-87-12, *University of Pisa, Dipartimento di Informatica*, June 1987.

[GG-87] Z. Galil and R. Giancarlo, “Data structures and algorithms for approximate string matching”, Research Report, *Columbia University, Computer Science Department*, 1987.

[LV-86] G.M. Landau and U. Vishkin, “Introducing efficient parallelism into approximate string matching and a new serial algorithm”, *Proc. 18th Annual ACM Symp. on Theory of Comput.*, Berkeley, CA, May 1986, 220-230.

[SK-83] D. Sankoff and J.B. Kruskal (Eds.), “*Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*”, Addison Wesley, Reading, MA, 1983.

[U-85] E. Ukkonen, “Finding approximate patterns in strings”, *J. of Algorithms* 6, 1985, 132-137.

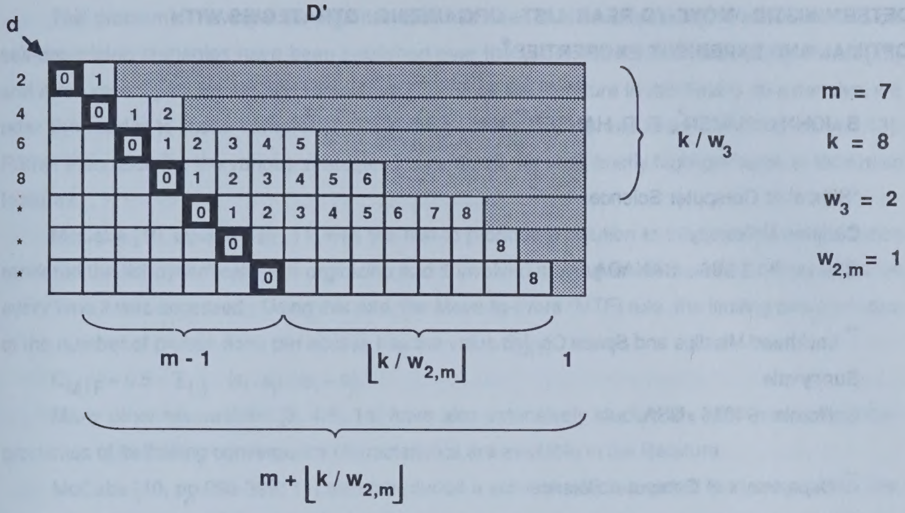

D'
d
2
4
6
8
*
*
*
0 1
0 1 2
0 1 2 3 4 5
0 1 2 3 4 5 6
0 1 2 3 4 5 6 7 8
0 8
0 8
k / w3
m = 7
k = 8
w3 = 2
w2,m = 1
m - 1
⌊k / w2,m⌋
1
m + ⌊k / w2,m⌋

Figure 1.

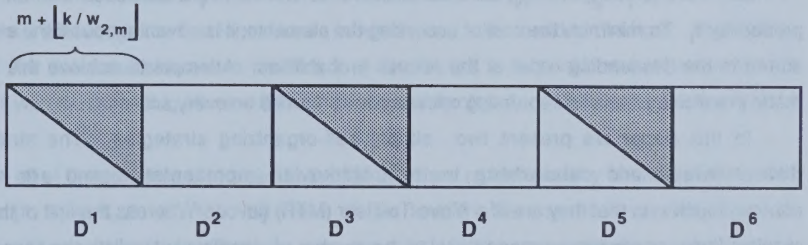

m + ⌊k / w2,m⌋
D1
D2
D3
D4
D5
D6

Figure 2.

DETERMINISTIC MOVE-TO-REAR LIST ORGANIZING STRATEGIES WITH OPTIMAL AND EXPEDIENT PROPERTIES†

B. JOHN OOMMEN*, E. R. HANSEN** and J. I. MUNRO***

*School of Computer Science
Carleton University
Ottawa : K1S 5B6 ; CANADA.

** Lockhead Missiles and Space Co. Inc.,
Sunnyvale
California : 94086 ; USA.

***Department of Computer Science
University of Waterloo
Waterloo N2L 3G1 ; CANADA.

ABSTRACT

Let $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$ be a list of elements in which R_i is accessed with an (unknown) probability s_i. To minimize the cost of accessing the elements, it is advantageous if the elements are sorted in the descending order of the access probabilities. Attempts to achieve this have been made in which a simple list reordering operation is performed on every access.

In this paper we present two simple self-organizing strategies. The strategies are **deterministic** and **absorbing** in their Markovian representation and are completely counter-intuitive, in that they are of a Move-To-Rear (MTR) flavour. Whereas the first of the schemes requires linear space (space proportional to the number of elements in the list), the second requires only constant space. We show that the former scheme is optimal, independent of the distribution of the access probabilities. By this we mean that although the list could converge to one of its N! configurations, by suitably performing the move-to-rear operation, the probability of converging to the right arrangement can be made as close to unity as desired. The second scheme requiring constant space is shown to be expedient. We conjecture its optimality.

I. INTRODUCTION

Suppose we are given a set of elements $\mathcal{R} = \{R_1, \ldots, R_N\}$. At every time instant one of these elements is accessed. Further, the element R_j is accessed with an unknown access probability s_j. We assume that the accesses are made independently. Whenever an element R_j is accessed, a sequential search is performed on the list. To minimize the cost of accessing, we intend to order the records in the descending order of their access probabilities.

This problem of having a file organize itself has been studied extensively. Various reviews on self-organizing strategies have been published over the years due to Gonnet et al [4], Rivest [13] and more recently by Hester and Hirschberg [7]. Since the literature in this field is so extensive, we refer the reader to these comprehensive surveys of the papers and results in the area [4,7,13]. Rather than describe the various strategies in any detail, we shall briefly highlight some of their main features.

McCabe [10, pp.398-399; 11] was the first to propose a solution to this problem. His solution rendered the list dynamically self-orginizing and it involved moving an element to the front of the list every time it was accessed. Using this rule, the Move-to-Front (MTF) rule, the limiting average value of the number of probes done per access has the value C_{MTF}, where,

$$C_{MTF} = 0.5 + \Sigma_{i,j} \ (s_i \cdot s_j) / (s_i + s_j).$$

Many other researchers [3, 4-6, 14] have also extensively studied the MTF rule and various properties of its limiting convergence characteristics are available in the literature.

McCabe [10, pp.298-399; 11] also introduced a scheme which is called the transposition rule. This rule requires that an accessed element is interchanged with its preceding element in the list, unless, of course, it is at the front of the list. Much literature is available on the transposition rule [1, 2, 10, 11, 13] but particular important is the work of Rivest [13] and Tenenbaum et al [1, 15] who extensively studied this rule and suggested its gereralizations - the Move-k-Ahead rule and the POS(k) rule. In the former, the accessed element is moved k-positions forward towards the front of the list unless it is found in the first k positions - in which case it is moved to the front. The POS(k) rule moves the accessed element to position k of the list if it is in positions k+1 through n. It transposes it with the preceding element if it is in positions 2 through k. If it is the first element it is left unchanged.

Almost all of the schemes discussed in the literature are represented by Markov chains which are ergodic [1, 3, 4, 6, 8, 13, 15]. By virtue of this fact, the list can be in any one of its N! configurations – even in the limit. Thus, for example, a list which was completely organized can be thoroughly disorganized by the MTF rule by a **single** request for the element which is accessed most infrequently. Observe that after this unfortunate occurrence, it will take a long time for the list to be organized again – i.e., for this element to dribble its way to the tail of the list.

If the access probabilities $\{s_i\}$ are time invariant, clearly, the ergodic Markovian representation of a list organizing scheme is an undesirable property. All the efforts taken for an element to learn its right position can be easily annulled by the occurrence of events (such as the access of the least frequently accessed element) which have finite probability. Indeed, for a particular scheme, even if the probability of the list being in the unique optimal configuration is large, the probability of it remaining in that configuration for any length of time **can** be arbitrarily close to zero.

As opposed to ergodic representations, Markovian behaviour can also be absorbing [9]. In an absorbing Markov chain, the chain converges to one of a (finite) set of absorbing barriers. The problem of computing the probability of converging to any **particular** barrier involves evaluating what is usually called the first passage probability [11].

The literature reports of only two absorbing list organizing schemes [7, 12] both of which are due to Oommen and Hansen, and have been described in [12]. Both the algorithms were essentially of a learning flavour and were **stochastic**, and in them, the elements of the list adaptively learned to find their place. The first algorithm in [12] is essentially a move-to--front algorithm with the exception that on the n^{th} access, the accessed element is moved to the front of the list with a probability f(n). This probability is systematically **decreased** every time an element is accessed. Ultimately on being accessed, each element tends to stay in the place where it is (as opposed to moving to the front of the list). This renders the Markovian representation of the procedure to be absorbing, as opposed to ergodic. The organization of the list gets "absorbed" into one of the N! orderings. It was shown in [12] that the scheme is expedient, i.e., if $s_i > s_j$, then the probability of absorption into an arrangement in which R_i precedes R_j is always greater than 0.5.

The second algorithm presented in [12] is far more powerful. In this case, the algorithm is a **move-to-rear** scheme in which the accessed element R_j is moved to the **rear** of the list with a probability q_j. This quantity q_j is progressively decremented every time **the element** is accessed. As before, ultimately there is no move operation performed on the list. Thus this scheme too is absorbing in its Markovian representation and so could converge into any one of its N! arrangements. However, in this case, it was shown that the probability of converging to the optimal arrangement can be made as close to **unity** as desired.

The latter technique requires more workspace than the traditionally used methods. However, apart from the latter algorithm being much more accurate than all the algorithms reported in the literature, it is also computationally more efficient. This is because the access of an element requires the update of **exactly** one probability (namely the probability q_j). Further, unlike the contemporary algorithms, since the list operations are essentially stochastic, a list operation is **not** necessarily performed on every access. Finally, since the Markovian representation of the scheme is absorbing, the number of list operations performed **asymptotically decreases to zero**. Furthermore, the superiority of the latter scheme of [12] to that of using counters has been proven in [12], primarily in terms of the number of updates and in terms of the probability of asymptotically performing move operations.

Apart from the schemes in [12] being absorbing, the most interesting property of the optimal one is that it is a **move-to-rear** scheme. This is indeed quite counter-intuitive .

In this paper, we shall present two new absorbing list organizing schemes. They are deterministic, of a **move-to-rear** flavour and are very simple to implement. Additionally, the number of move operations performed in both the schemes are **exactly** equal to the number of elements in the list, primarily because a move operation is performed on an element **exactly** once. Whereas the first of the schemes requires linear space, the second requires only constant space (i.e., space independent of the number of elements in the list). The most powerful property of the former scheme is that it is optimal, in that the probability of being absorbed into the optimal configuration can be made as close to unity as desired. The second scheme requiring constant space is shown to be expedient. In other words we shall show that if $s_i > s_j$, then the probability of being absorbed into an arrangement in which R_i precedes R_j is greater than 0.5. We conjecture its optimality.

Although both of the schemes described in this paper are conceptually Move-to-Rear schemes, the potential drawbacks that a MTR scheme could possess are absent in both of them primarily because they can be implemented very conveniently by introducing **a single** additional pointer. In a pure MTR scheme the elements which are accessed very infrequently would linger at the front of the list till they are themselves to the rear of the list. These infrequently accessed elements could thus potentially prove to be a drag on the entire access strategy. However, this can be easily overcome by maintaining two list pointers, the first pointer always pointing to the front of the original list, and the second pointer pointing to an interior element of the list. This interior element is indeed the element which was **first** moved to the rear. The details of this implementation and the access strategy will be discussed later.

II. A DETERMINISTIC LINEAR SPACE MOVE-TO-REAR STRATEGY

We shall now present a deterministic absorbing list organizing strategy and show that it is asymptotically optimal. The idea behind the scheme is extremely straightforward. Let T be any fixed integer. Associated with every element R_i is an integer memory location x_i which is initially set to zero. Whenever the element R_i is accessed x_i is incremented, but **no** list reorganization is performed. Whenever the value of x_i is exactly T, the element R_i is moved to the **rear** of the list. Subsequently, the element R_i is neither moved nor its associated location x_i updated. This describes the list organization strategy completely. This is algorithmically given below.

ALGORITHM Deterministic-MTR-LSpace

Input : A sequence of accesses on the list $R = \{R_1, R_2, \ldots, R_N\}$. R_i is accessed with an unknown access probability s_i. This is read in using procedure ReadInput.

Output : A reorganized list.

Memory Requirements :

(i) An integer memory location x_i associated with every R_i, (ii) A boolean variable M_i associated with R_i. M_i indicates if R_i has already been moved and (iii) An integer constant T.

```
Method :  For i = 1 to N do
              $x_i = 0$
              $M_i$ = False
          EndFor
          Repeat
              ReadInput ($R_i$)
              If Not ($M_i$) then
                  $x_i = x_i + 1$
                  If ($x_i = T$) then
                      Move-to-Rear ($R_i$)
                      $M_i$ = True
                  EndIf
              EndIf
          Forever
End Algorithm Deterministic-MTR-LSpace
```

We shall now allude to the properties of the above strategy.

Theorem I

The deterministic Move-to-Rear Scheme described above is absorbing. Furthermore, the total number of list reorganizing operations done is exactly N.

Proof : The theorem is proved elsewhere [17]. • • •

The power of the above scheme lies primarily in the fact that it is not just the **expected** number of move operations which is finite. This was, of course, true of the stochastic schemes described in [12]. In this case however, the number of moves performed, **not being a random variable**, is exactly equal to the size of the list.

The following results concerning the probability of being absorbed in the optimal configuration is more powerful. We shall show that the scheme is asymptotically optimal, i.e., the probability of being absorbed in the optimal configuration can be made as close to unity as desired.

Theorem II

For any distinct indices u, v ε {1, 2, ..., N}, let $\xi_{u,v}$ be the event that either R_u or R_v is accessed. Further, let ${}_uP_v$ be the **conditional** probability that R_u ultimately precedes R_v given $\xi_{u,v}$. Then,

(i) ${}_uP_v$ can be made as to close to unity as desired if $s_u > s_v$ and,

(ii) ${}_uP_v$ is exactly 0.5 if $s_u = s_v$.

Proof : The proof of the theorem is quite involved and is given in [17]. • • •

The final theorem regarding the asymptotic optimality of the scheme now follows directly.

Theorem III

Let $R = \{R_1, R_2, \ldots, R_N\}$ be the list of elements with distinct access probabilities $\{s_1, \ldots, s_N\}$. Then, the probability of the list converging into the optimal configuration can be made as close to unity as desired.

Proof : The proof of the theorem is given in [17]. • • •

III. A DETERMINISTIC CONSTANT SPACE MOVE-TO-REAR STRATEGY

We now present a deterministic absorbing list organizing strategy that requires only **constant** space and show that it is expedient. In other words we shall show that for this scheme, if $s_i > s_j$, then the probability of being absorbed into an arrangement in which R_i precedes R_j is greater than 0.5. We conjecture that the optimality of this scheme.

Let T be any fixed integer and Z be a memory location having two integer fields, z_1 and Z_2. Whenever the element R_i is accessed Z_1 is set to i. Furthermore, if Z_1 was previously set to i, Z_2 is incremented. Otherwise the value of Z_2 is reset to zero. Whenever the value of Z_2 is exactly T, the element R_i (i.e., the element whose index is stored in Z_1) is moved to the **rear** of the list. Subsequently, the element R_i is neither moved nor its associated location x_i updated. This describes the list organization strategy completely. Observe that essentially all that this scheme does is that it moves a record R_i to the rear of the list after it has been accessed T **consecutive** times.

For the sake of completeness, we shall algorithmically describe the above process.

ALGORITHM Deterministic-MTR-CSpace

Input : A sequence of accesses on the list $R = \{R_1, R_2, \ldots, R_N\}$. R_i is accessed with an unknown access probability s_i. This is read in using procedure ReadInput.

Output : A reorganized list.

Memory Requirements :

(i) An memory location Z with two integer fields Z_1 and Z_2. Z_1 stores the index of the last accessed record, and Z_2 counts the number of times the record whose index is Z_1 has been consecutively accessed.

(ii) A boolean variable M_i associated with R_i. M_i indicates if a MTR (i.e., a list operation) has been performed on R_i. Although this implies that the algorithm requires linear space (since a memory location M_i has to be maintained for ever record R_i), we shall later show that by the introduction of a **single** extra pointer the implementation can be rendered to require **only** constant space. However, the pseudo-code is more understandable if we assume that every record has a single additional **bit** of information storing whether it has been moved or not.

(iii) An integer constant T.

```
Method :  For i = 1 to N do
                    $M_i$ = False
          EndFor
          $Z_1 = -\infty$
          Repeat
                    ReadInput ($R_i$)
                    If Not ($M_i$) then
                              If ($Z_1 = i$) then
                                        $Z_2 = Z_2 + 1$
                              Else
                                        $Z_2 = 1$
                                        $Z_1 = i$
                              Endif
                              If ($Z_2 = T$) then
                                        Move-to-Rear ($R_i$)
                                        $M_i$ = True
                              EndIf
                    EndIf
          Forever
End  Algorithm  Deterministic-MTR-CSpace
```

We shall now state the properties of the above strategy.

Theorem IV

The constant space deterministic MTR strategy described by Algorithm Deterministic-MTR-CSpace above is expedient.

Proof : The theorem is proved in [17]. • • •

Apart from the above scheme being expedient we also believe the following conjecture.

Conjecture I

The constant space deterministic MTR strategy described by Algorithm Deterministic-MTR-CSpace above is asymptotically optimal. • • •

For any distinct indices u, $v \varepsilon \{1, 2, \ldots, N\}$, let $\xi_{u,v}$ be the event that either R_u or R_v is accessed. Further, let ${}_uP_v$ be the **conditional** probability that R_u ultimately precedes R_v given $\xi_{u,v}$. Then, the proof of the conjecture will involve proving that if T is increased indefinitely,

(i) ${}_uP_v$ is can be made as close to unity as desired if $s_u > s_v$, and,

(ii) ${}_uP_v$ is exactly 0.5 if $s_u = s_v$.

The proof of this conjecture is by no means trivial. It involves the solution of a random walk, in which the random "walker" can make single step traversals of length up to and including T/2. We hope to prove this conjecture in the near future. Should this conjecture be true, the constant space deterministic MTR strategy will be the **ideal** list organizing strategy. The strategy requires only a constant amount of extra memory, performs exactly one list reordering operation per record and converges in an **absorbing** (as opposed to ergodic) mode to a final ordering. Furthermore, should this conjecture be true, the probability that this final ordering is the **optimal** ordering can be made as close to unity as desired.

IV. IMPLEMENTATION DETAILS

It is interesting to understand how the MTR schemes discussed in this paper may be actually implemented. To explain the implementation details, we shall merely specify these details for the case of the linear space scheme described in Section II. The principles are analogous for the constant space scheme described in Section III.

Consider the linear space scheme described in Section II. Observe that initially the schemes **actually** moves the most frequently accessed elements to the rear of the list. Thus, the elements which are accessed very infrequently would linger at the front of the list till they are accessed T times, and thus prove to be a drag on the access strategy. However, this can be easily overcome by maintaining two list pointers FrontOld and FrontNew. The first pointer, FrontOld always points to the front of the original list, and the second pointer, FrontNew, points to the element in the list which was **first** moved to the rear.

When an access on R_i is made, the list pointed to by FrontNew is first searched, and if the element is found, **neither** a counter update **nor** a list reorganization operation is executed. However, if the element R_i is not found in the list pointed to by FrontNew, a search is made by traversing the list which is pointed at by FrontOld. The latter traversal terminates at FrontNew and **may** result in a list reorganization.

The implementation of the scheme is shown in Figure I for a list of five elements, namely the list $R = \{R_1, R_2, ..., R_5\}$ in which $s_1 > s_2 > s_3 > s_4 > s_5$. In Figure I we have considered a case when the variables $\{ x_i \}$ attain the value T in the order (1,2,3,4,5). Initially, on R_1 being accessed T times, it is moved to the rear of the list. Subsequently, on R_2 being accessed T times, **it** is moved to the rear of the list. As time proceeds, whenever $x_3 = T$, R_3 is moved to the rear of the list. However, to catalyze the accessing process, the list pointers FrontOld and FrontNew are maintained as described earlier. Note that ultimately, FrontOld and FrontNew point to the reorganized list.

IV. CONCLUSIONS

We have considered a list of elements $R = \{R_1, ..., R_N\}$ in which the element R_i is accessed with an unknown probability s_i. Two deterministic list organizing schemes have been proposed. The schemes have an absorbing Markovian representation, and he number of move operations performed in both the schemes is **exactly** equal to the number of elements in the list. Whereas the first of the schemes requires linear space (space proportional to the number of elements in the list), the second requires only constant space. We have shown that the former scheme is optimal. In other words, the probability of being absorbed into the optimal configuration can be made as close to unity as desired. The second scheme requiring constant space is shown to be expedient. We conjecture its optimality.

ACKNOWLEDGEMENTS

The authors are very grateful to David Ng who helped in preparing the final manuscript and the drawings. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Arnow, D.M. and Tenebaum, A.M., "An Investigation of the Move-Ahead-k Rules", Congressus Numerantium, Proc. of the Thirteenth Southeastern Conference on Combinatorics, Graph Theory and Computing, Florida, February 1982, pp. 47-65.

[2] Bitner, J.R., "Heurishtics That Dynamically Organize Data Structures", SIAM J. Comput., Vol.8, 1979, pp. 82-110.

[3] Burville, P.J. and Kingman, J.F.C., "On a Model for Storage and Search", J. Appl. Probability, Vol.10, 1973, pp. 697-701.

[4] Gonnet, G.H., Munro, J.I. and Suwanda, H., "Exegesis of Self Organizing Linear Search", SIAM J. Comput., Vol.10, 1981, pp.613-637.

[5] Hansen, E.R., A Table of Series and Products, Prentice Hall, Englewood Cliffs, N.J., 1975.

[6] Hendricks, W.J., "An Extension of a Theorem Concerning an Interesting Markov Chain", J. App. Probability, Vol.10, 1973, pp.231-233.

[7] Hester, J.H. and Hirschberg, D.S., "Self-Organizing Linear Search", ACM Computing Surveys, Vol. 17, 1985, pp.295-311.

[8] Kan, Y.C. and Ross, S.M., "Optimal List Order Under Partial Memory Constraints", J. App. Probability, Vol.17, 1980, pp. 1004-1015.

[9] Karlin, S. and Taylor, H.M., " A First Course in Stochastic Processes", Academic Press, 1975.

[10] Knuth, D.E., "The Art of Computer Programming, Vol.3, Sorting and Searching", Addison-Wesley, Reading, MA., 1973.

[11] McCabe, J., "On Serial Files With Relocatable Records", Operations Research, Vol.12, 1965, pp.609-618. 43.

[12] Oommen, B.J., and Hansen E.R., "List Organizing Strategies using Stochastic Move-to-Front and Stochastic Move-to-Rear Operations", SIAM Journal of Computing, Vol. 16, No. 4, August 1987.

[13] Rivest, R.L., "On Self-Organizing Sequential Search Heuristics", Comm. ACM, Vol.19, 1976, pp.63-67.

[14] Sleator, D. and Tarjan, R., "Amortized Efficiency of List Update Rules", Proc. of the Sixteenth Annual ACM Symposium on Theory of Computing, April 1984, pp. 488-492.

[15] Tenenbaum, A.M. and Nemes, R.M., "Two Spectra of Self-Organizing Sequential Search Algorithms", SIAM J. Comput., Vol.11, 1982, pp-557-566.

[16] Abramowitz, M. and Stegun I.A., Handbook of Mathematical Functions, National Bureau of Standards, Dover Publications, Inc., New York, 1970.

[17] Oommen, B.J., Hansen, E.R. and Munro, J.I., "Deterministic Optimal and Expedient Move-to-Rear List Organizing Strategies". Submitted for Publication.

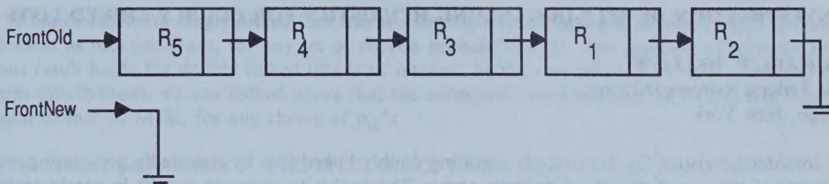

FrontOld
R_5
R_4
R_3
R_1
R_2
FrontNew

Figure I (a)

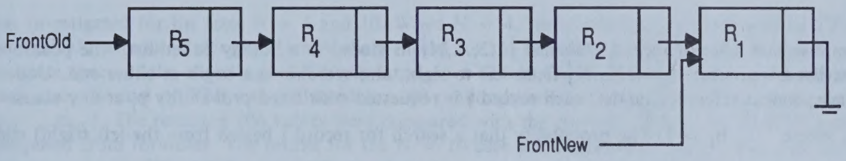

FrontOld
R_5
R_4
R_3
R_2
R_1
FrontNew

Figure I (b)

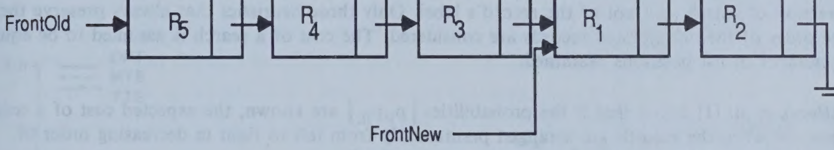

FrontOld
R_5
R_4
R_3
R_1
R_2
FrontNew

Figure I (c)

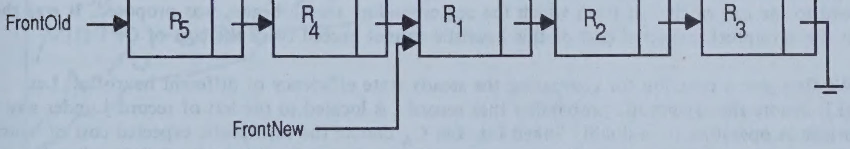

FrontOld
R_5
R_4
R_1
R_2
R_3
FrontNew

Figure I (d)

Figure I : Implementation of the Move-to-Rear Scheme. FrontOld points to the original Head of the list and FrontNew points to the element which was first moved to the rear. The list is traversed from FrontNew, and if the element sought for is not obtained, it is then traversed from FrontOld.

AN INVESTIGATION OF SELF-ORGANIZING HEURISTICS FOR DOUBLY LINKED LISTS

MICHAEL E. BELLOW
IBM Federal Systems Division
Owego, New York

We consider heuristics for dynamically updating doubly linked lists by repeatedly performing rearrangements of the stored records as requests arrive. The goal is to maintain the list in nearly optimal order with respect to average retrieval cost. Because of their bidirectional traversal capability, doubly linked lists have important applications in computing. Self-organizing heuristics for singly linked lists have received considerable attention, but not much work has been done in this area for doubly linked lists.

Suppose that a set of records, labelled $\{1,2,\ldots,N\}$, is stored in a doubly linked list. The positions of the list are numbered $\{1,2,\ldots,N\}$ from left to right, and searches can begin at either end. Under the independent reference model, each record j is requested with fixed probability p_j at any access time, where $\sum_{j=1}^{N} p_j = 1$. The probability that a search for record j begins from the left (right) end of the list is given by p_{jL} (p_{jR}), where $p_{jL} + p_{jR} = 1$, for each j.

The starting point of a given search is assumed to be selected independently of the request for the record being sought, and independently of the starting points of all previous searches. Once a requested record has been retrieved, the list can be restructured according to a permutation rule (heuristic). The specific permutation applied is a function of the position where the record was found and the direction of search, but not of the record's label. Only those heuristics that always preserve the relative order of the unrequested records are considered. The cost of a search is assumed to be equal to the number of list positions examined.

Matthews et al. [1] found that if the probabilities $\{p_j, p_{jL}\}$ are known, the expected cost of a search is minimized when the records are arranged permanently from left to right in decreasing order of $\{p_j(p_{jL} - ½)\}$. This is known as the optimal ordering (OPT).

When the probabilities are unknown, the objective is to find a self-organizing heuristic with asymptotic expected cost as close as possible to that of OPT. If the p_{jL}'s are all equal, the problem reduces to consideration of heuristics for singly linked lists. For the situation where at least one of the p_{jL}'s is different from the others, a heuristic denoted move-to-end (MTE), which always moves the requested record to the end of the list from which the corresponding search began, was proposed. It was shown that the asymptotic expected cost of this heuristic cannot exceed twice the cost of OPT [1].

We first give a criterion for comparing the steady state efficiency of different heuristics. Let $b_A(i,j)$ denote the asymptotic probability that record i is located to the left of record j under any heuristic A operating on a doubly linked list. Let C_A denote the asymptotic expected cost of heuristic A. We can show that if A_1 and A_2 are two heuristics such that $b_{A_1}(i,j) \geq b_{A_2}(i,j)$ for all pairs of records (i,j) such that $p_i(p_{iL} - ½) \geq p_j(p_{jL} - ½)$, then $C_{A_1} \leq C_{A_2}$.

This criterion can be used to prove that the move-to-end heuristic outperforms the move-to-left heuristic (MTL) if the relative order of $\{p_j p_{jL}\}$ is the same as that of $\{p_j(p_{jL} - ½)\}$. The MTL heuristic always moves the requested record to the left end of the list; it is a version of the move-to-front heuristic commonly used to update singly linked lists. By symmetry, the analogous result holds for a move-to-right scheme.

We next consider a heuristic known as transpose-toward-end (TTE) [2]. This is an analogue of the transpose heuristic for singly linked lists. The TTE scheme moves the requested record by one position toward the end of the list from which the search began. Thus, if the requested record is found in position j by a left-to-right search, it is moved to position $\max(j-1,1)$; if it is found in position j by a right-to-left search, then it is moved to position $\min(j+1,N)$.

A major result for singly linked lists was the finding that transpose is asymptotically at least as efficient as move-to-front, for any set of request probabilities [3]. The question of whether an analogous result holds for doubly linked lists is of interest. In the case where $p_j = 1/N$, $j=1,\ldots,N$ (uniform distribution), we can indeed prove that the asymptotic expected cost of TTE is less than or equal to that of MTE, for any choice of p_{jL}'s.

The relative performance of TTE, MTE, and OPT when the p_j's follow Zipf's law:

$$p_j = \frac{1}{j \sum_{i=1}^{N} 1/i}, \quad j = 1,\ldots,N$$

was investigated for list sizes N = 4 and 10. When N = 4, the asymptotic expected cost of TTE can be explicitly computed using a Markov chain model for the changing list configuration. For N = 10, the state space of the chain is prohibitively large, so a Monte Carlo simulation was used instead. In each case, the cost of TTE was evaluated for 100 randomly selected values of the parameter vector $(p_{1L},\ldots,p_{NL})$. The resulting 200 values were compared with the corresponding costs of MTE and OPT computed from formulas. The results for the N = 10 case are displayed in Fig. 1. The computed average costs of TTE, MTE, and OPT are plotted as a function of run, with the runs ordered by increasing OPT cost. It was found that TTE outperformed MTE on all 100 runs in both cases, and the average cost of TTE never exceeded a factor of 1.14 of the optimal cost. This is strong evidence in favor of using the TTE heuristic to update doubly linked lists when requests can be assumed independent.

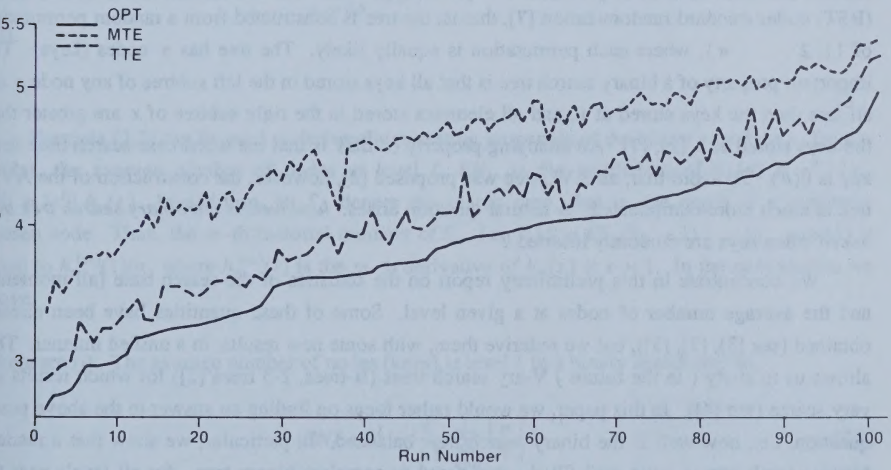


Figure 1. Results for N = 10

References

1. D. Matthews, D. Rotem, and E. Bretholz, "Self-organizing doubly linked lists", *International Journal of Computer Mathematics,* vol. 8, pp. 99-106, 1980.

2. M. Bellow, "Performance of Self-Organizing Sequential Search Heuristics under Stochastic Reference Models", Ph.D. thesis, Carnegie-Mellon University, 1984.

3. R. Rivest, "On self-organizing sequential search heuristics", *Communications of the ACM,* vol. 19, no. 2, pp. 63-67, 1976.

HOW WELL ARE BINARY SEARCH TREES BALANCED ?

WOJCIECH SZPANKOWSKI* and VERNON REGO*

Department of Computer Science
Purdue University
W. Lafayette, IN 47907

Abstract. *We consider a binary search tree with n nodes randomly inserted in the tree. In such an environment we investigate similarities between binary search trees and an equivalent complete tree with n nodes, that is, how closely the shape of a complete tree approximates a random binary search tree. To answer this question we study the average number of nodes in a binary search tree at any level, and all moments of the search time for a given (randomly selected) key. In particular, we prove that for all levels l such that* $l = o(\ln n)$ *the difference between the number of nodes in a complete tree and a binary search tree is* $0(\ln^{l-1} n/n)$. *In addition, we prove that the average and the variance of the search time is asymptotically* $2\ln n$ *compared to* $\log_2 n$ *and* $0(1)$ *respectively for complete trees and digital search trees [7], [9].*

1. INTRODUCTION

The binary search tree is a well-studied data structure. We investigate the binary search tree (BST) under standard randomization [7], that is, the tree is constructed from a random permutation of $\{1, 2, \ldots, n\}$, where each permutation is equally likely. The tree has n nodes (keys). The important property of a binary search tree is that all keys stored in the left subtree of any node x are all less than the keys stored at x, and all elements stored in the right subtree of x are greater than the keys stored at x [2], [7]. An annoying property of BST is that the worst case search time for a key is $0(n)$. To avoid that, an AVL-tree was proposed [2], however, the construction of the AVL-tree is much more complicated. A natural question arises: *how well is the binary search tree balanced* when keys are randomly inserted ?

We concentrate in this preliminary report on the statistics of the search time (all moments) and the average number of nodes at a given level. Some of these quantities have been already obtained (see [3], [7], [8]), but we rederive them, with some new results, in a unified manner. This allows us to study (in the future) V-ary search trees (B-trees, 2-3 trees [2]) for which results are very scarce (see [4]). In this paper, we would rather focus on finding an answer to the above posed question, i.e., how well is the binary search tree balanced. In particular, we show that a random binary search tree is quite well filled - compared to complete binary tree - for all levels near the root. More precisely, for all levels l such that $l = o(\ln n)$ the average wasted space is $0(\ln^{l-1} n/n)$. On the other hand, the average and the variance of the search time is asymptotically $2 \ln n$ compared to $\log_2 n$ and $0(1)$ respectively for complete binary trees and digital search trees [7], [9], [10].

---

* This research for both authors was supported in part by the NSF under grant NCR-8702115.

2. SUMMARY OF MAIN RESULTS

We consider a binary search tree with n keys (nodes), and we adopt the usual assumption that all n permutations of the input are equally likely. In such an environment, let us define a generating function $h_n(z)$ with the coefficient at z^k, denoted $[z^k]\, h_n(z)$, being the expected number of nodes at level k in the family of random binary search trees built from n records. Under the assumption that all permutations of the input are equally likely, we easily obtain the recurrence equation for $h_n(z)$. Namely

$$\begin{aligned} &h_0(z) = 0, \quad h_1(z) = 1 \\ &h_n(z) = \frac{2z}{n} \sum_{k=0}^{n-1} h_k(z) + 1 \qquad n \geq 2 \end{aligned} \tag{2.1}$$

In the next section, we prove that (2.1) has the following solution

Proposition. Let $G_n(z) = z(z+1)(z+2) \ldots (z+n-1)$. Then (2.1) possesses the solution

$$h_n(z) = \frac{G_n(2z)}{n!(2z-1)} - \frac{1}{2z-1} \tag{2.2}$$

for all $z \geq 0$. (Note that for $z = 0.5$ $\lim_{z \to 0.5} h_n(z) = H_n - 1$ where H_n is the n-th harmonic number [7]).

□

Formula (2.2) can be used to derive all interesting properties of the binary search tree. In particular, the average number of nodes at level l, EN_l, is the coefficient of $h_n(z)$ at z^k, i.e., $EN_l = [z^l]\, h_n(z)$. In addition, let S_n denote the search time, that is, the depth of a randomly chosen node. Then, the m-th factorial moment of S_n, i.e., $ES_n^{\underline{m}} = ES_n(S_n - 1) \ldots (S_n - m+1)$ is equal to $h_n^{(m)}(1)/n$, where $h_n^{(m)}(1)$ is the m-th derivative of $h_n(z)$ at $z = 1$. In the next section we prove

Theorem (i). The average number of nodes (keys) at level l in a binary search tree is

$$EN_l = 2^l - \frac{2^l}{n!} \sum_{k=0}^{l} \begin{bmatrix} n \\ l \end{bmatrix} \tag{2.3}$$

where $\begin{bmatrix} n \\ l \end{bmatrix}$ is the Stirling number of the first kind [6]. If $l = 0(\ln n)$ then

$$EN_l \sim 2^l - \frac{2^l}{(l-1)!} \frac{(\gamma + \ln n)^{l-1}}{n} = 2^l - 0(\ln^{l-1} n/n) \tag{2.4}$$

(ii) The average search time, ES_n, satisfies

$$ES_n = 2[1 + \frac{1}{n}]H_{n+1} - 4 + 0(n^{-1}) \sim 2 \ln n \tag{2.5}$$

(iii) The variance of the search time is

$$\text{var } S_n = 2\left[1+\frac{1}{n}\right]H_{n+1} - 4\left[1+\frac{1}{n}\right](H_{n+1}^2/n - H_{n+1}^{(2)}) + 4 \sim 2 \ln n \tag{2.6}$$

(iv) The m-th moment of S_n satisfies

$$ES_n^m \sim 2^m \, H_{n+1}^m \sim 2^n \ln^n n \tag{2.7}$$

□

As a simple consequence of the Theorem, we obtain

Corollary (i). The probability $Pr\{S_n = l\}$ becomes

$$Pr\{S_n = l\} \frac{2^l}{n\, n!} \sum_{k=l+1}^{n} \begin{bmatrix} n \\ k \end{bmatrix} \tag{2.8}$$

(ii) The average number of vacancies at level l, EV_l, that is, the number of positions where the n+1-st node can be inserted at level l, can be computed as:

$$EV_l = \frac{2^l}{n!} \begin{bmatrix} n \\ l \end{bmatrix} \tag{2.9}$$

If $l = o(\ln n)$, then

$$EV_l \sim \frac{2^l(\gamma + \ln n)^{l-1}}{n} \tag{2.10}$$

□

Both the Theorem and Corollary characterize the balancing property of the binary search tree. Let us start our discussion with the consequences of Eq.(2.3). It is well known that a complete binary tree has 2^l nodes at the l-th level, if l is not the last level of the tree. Therefore, formula (2.3) indicates that the number of wasted space at level l, is determined by the second term of (2.3). In particular, for $l = o(\ln n)$, that is, at levels near the root, the wasted space is $0(\ln^{l-1} n/n)$ as (2.4) suggests. This indicates that a binary search tree is fairly well balanced. However, investigation of the search time shakes that assertion. The average search time in BST is asymptotically equal to $2 \ln n$ while in a complete binary tree, the equivalent quantity is $\log_2 n$. Moreover, the variance of the search time increases as $2 \ln n$. Note that in all three possible digital search trees (e.g., radix search trie, Patricia trie and digital search tree) the successful search has the mean equal to $\log_2 n$ and the variance becomes $0(1)$ [9], [10]. Moreover, the m-th moment of the search time is asymptotically equal to $\log_2^m n$ for digital search trees, and $2^m \ln^n n$ for binary search trees.

3. ANALYSIS

In this section, we prove our main results discussed before, and we also point out how our analysis can be extended to study V-ary binary search trees.

The ''heart'' of our analysis is the Proposition. To prove Eq.(2.2) we can use elementary calculus to solve the recurrence (2.1). However, we adopt a different approach, since it can be easily generalized to V-ary binary search trees, which is our ultimate goal. Let us define a new generating function, $h(z,v)$ as

$$h(z,v) = \sum_{h=0}^{\infty} h_n(z)u^n \tag{3.1}$$

Then multiplying (2.1) by u^{n-1} and summing up, one immediately proves

$$\frac{\partial h(z,u)}{\partial u} = 2z\,\frac{h(z,u)}{1-u} + \frac{1}{(1-u)^2} \tag{3.2}$$

If z is fixed, then (3.2) can be considered as an ordinary differential equations with respect to u. It is easy to see that (3.2) falls into Euler's nonhomogeneous differential equation. Hence, the solution of the homogeneous equation of (3.2) is $(1-u)^{-2z}/(2z-1)$ [5], while the particular solution is $1/[(2z-1)(1-u)]$. That is,

$$h(z,u) = \frac{(1-u)^{-2z}}{(2z-1)} - \frac{1}{(2z-1)(1-u)} \tag{3.3}$$

To find $h_n(z)$, we need to identify the coefficient of u^n of $(1-u)^{-2z}$. But

$$(1-u)^{-2z} = \sum_{n=0}^{\infty} 2z(2z+1)\,.\,.\,.\,(2z+n-1)u^n = \sum_{n=0}^{\infty} G_n(2z)u^n$$

$$\frac{1}{1-u} = \sum_{n=0}^{\infty} u^n$$

where $G_n(z) = z(z+1) \cdots (2+n-1)$. This proves the Proposition.

Before we prove the Theorem, we show first how the method proposed above can be extended to compute the generating function, $h_n(z)$, for a V-ary binary search tree (for definition see [2], [4], [7]). It can be proved that in this case the recurrence for $h_n(z)$ satisfies

$$\begin{aligned} h_n(z) &= n \qquad n \le V-1 \\ h_n(z) &= \frac{zV}{\left[\begin{matrix} n \\ V-1 \end{matrix}\right]} \sum_{k=0}^{n-V+1} h_k(z) + V - 1 \end{aligned} \tag{3.4}$$

Let $h(z,u)$ be the generating function of $h_n(z)$ as defined in (3.1). Fixing z, and multiplying (3.4) by z^{n-V+1} one proves

$$\frac{d^{V-1}\,h(z,u)}{du^{V-1}} = V!\,z\,\frac{h(z,u)}{(1-u)^{V-1}} + \frac{(V-1)(V-1)!}{(1-u)^V} \tag{3.5}$$

The solution of (3.5) is available since the homogeneous part of equation (3.5) is again Euler's differential equation. Let us define $\alpha_i(z)$, $i = 1, 2, \ldots, V-1$ as the roots of the following equation

$$\alpha(z)[\alpha(z) - 1] \ldots [\alpha(z) - V+1] = (-1)^{V-1} V!\, z \tag{3.6}$$

Then the solution of the homogeneous differential equation (3.5) is

$$h(z,u) = \sum_{i=1}^{V-1} A_i(z)(1-u)^{\alpha_i(z)} \tag{3.7}$$

where $A_i(z)$ can be obtained from the initial conditions. A particular solution for (3.5) is $(V-1)/[(1-Vz)(1-u)]$. Therefore, a general solution of (3.5) becomes

$$h(z,u) = \frac{V-1}{(1-Vz)(1-u)} + \sum_{i=1}^{V-1} A_i(z)(1-u)^{\alpha_i(z)} \tag{3.8}$$

and identifying coefficients of $h(z,u)$ at u^n one proves

$$h_n(z) = \frac{V-1}{1-Vz} + \frac{1}{n!} \sum_{i=1}^{V-1} A_i(z)[\alpha_i(z)]_{(n)} \tag{3.9}$$

where $(\alpha)_n = \alpha(\alpha-1) \ldots (\alpha - 1+n)$.

Now we return to binary search trees and prove the Theorem. Note that $G_n(z) = z(z+1) \ldots (z+n-1)$ is the generating function of the Stirling number of the first kind, that is, $G_n(2z) = \sum_{k=0}^{n} \left[{n \atop k} \right] 2^k z^k$ [6, p.90]. Hence $[z^l]G_n(2z)/(2z-1) = -2^l \sum_{k=0}^{l} \left[{n \atop k} \right]$. Also $1/(2z-1) = -\sum_{k=0}^{\infty} 2^k z^k$. This proves Theorem (i), Eq. (2.3). Eq.(2.4) follows from the fact that $\left[{n \atop k} \right] \sim (n-1)(\gamma + \ln n)^{k-1}/(k-1)!$ for $k = o(\ln n)$ [1].

To prove the assertions on the search time (more precisely: the time required for MEMBER operations) we must compute the derivatives of $h_n(z)$. Let $h_n^{(m)}(1)$ be the m-th derivatives of $h_n(z)$ at $z = 1$. Then $ES_n^m = h_n^{(m)}(1)/n$. Using (2.2) we have to know the derivatives of $G_n(2z)$. Let $l(z) = \ln G_n(z) = \sum_{k=0}^{n-1} \ln(2z+k)$. It is easy to show that $l^{(m)}(1) = (-1)^{m+1} 2^m (m-1)![H_{m+1}^{(m)} - 1]$, where $H_n^{(m)}$ is the m-th generalization of the harmonic numbers [6, p.74]. Let also $g(z) = G_n(2z)$. Then

$$g^{(m)}(1) = \sum_{i=0}^{m-1} \left(\begin{matrix} k-1 \\ i \end{matrix} \right] g^{(i)}(1)\, l^{(m-i)}(1) \tag{3.10}$$

For example, $g(1) = (n+1)!$, $g'(1) = 2(H_{n+1} - 1)(n+1)!$, $g''(1) = 4(n+1)!\,[(H_{n+1} - 1)^2 - H_{n+1}^{(2)} + 1]$, etc. Finally, let $a(z) = (2z-1)^{-1}$. Naturally, $a^{(m)}(1) = (-1)^m 2^m m!$, and (2.2) with the above imply

$$h^{(m)}(1) = \frac{1}{n!} \sum_{l=0}^{n} \left[{m \atop l} \right] g^{(l)}(1)\, a^{(m-1)}(1) - a^{(m)}(1) \tag{3.11}$$

Substituting appropriate formulas into (3.11), one proves Theorem (iii)–(iv).

At last, we need to prove the Corollary. Formula (2.8) follows immediately from $Pr\{S_n = l\} = [z^l]h_n(z)/n = EN_l/n$ and Eq.(2.3). To prove (2.9) note that the average number of vacancies, V_l and the average number of nodes at level l, EN_l, satisfy a simple recurrence

$$EN_{l+1} + V_{l+1} = 2EN_l \tag{3.12}$$

Using (2.3) one proves (2.9).

References

[1] Abramovitz, M., Stegun, I., Handbook of mathematical functions, Dover (1965).

[2] Aho, A., Hopcroft, J., Ullmann, J., Data structures and algorithms, Addison-Wesley, Reading {1983).

[3]. Brown, G., Shubert, B., On random binary trees, *Math. Op. Res., 9 (1984), pp. 54–65.*

[4]. Mahmoud, H., On the average internal path length of m-ary search trees, *Acta Informatica,* 23 (1986), pp. 111–117.

[5]. Murphy, G., Ordinary differential equations and their solutions, D. Van Nostrad Company, Princeton (1960).

[6]. Knuth, D.E., The art of computer programming, vol. I., Addison Wesley, Reading (1963).

[7]. Knuth, D.E., The art of computer programming, vol. III, Addison Wesley, Reading (1973).

[8]. Louchard, G., Exact and asymptotic distributions in digital and binary search trees, preprint.

[9]. Szpankowski, W., Some results on V-ary asymmetric tries, to appear in *Journal of Algorithms.*

[10]. Szpankowski, W., Two problems on the average complexity of digital trees, *Proc. of PERFORMANCE'87*, Brussels (1987).

Necessary and Sufficient Conditions for Stability of Polytopes of Matrices through Tests of Lower Dimensional Subsets

Christopher L. DeMarco
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53703

ABSTRACT

Recent results on stability of families of polynomials whose coefficients lie in a polytope have indicated that the family is stable if and only if all members of the one dimensional subsets composing the "edges" of the polytope are stable. One might reasonably ask if a similar result holds for families of matrices whose elements lie in a polytope. This paper indicates that a weaker version of the "edge result" does indeed hold for nxn real matrices; specifically, all members of the polytope are stable if and only if all matrices in a specified collection of 2n dimensional affine subsets of the boundary of the polytope are stable.

I. BACKGROUND AND INTRODUCTION

The results reported in [1] indicate that given a family of polynomials described as a polytope in the space of coefficients, a necessary and sufficient condition for stability of the family is that all polynomials corresponding to convex combinations of pairs of extremal points must be stable. Indeed, [1] goes further to show that it is sufficient to check "edges;" i.e. convex combinations of pairs of extremal points that lie entirely on the boundary of the polytope. This result is significant in the sense that determining the stability of a polytope is reduced to determining stability of a sequence of one dimensional subsets. However, given an engineering problem modeling a physical system in which some parameters are imprecisely known, the uncertainty in the set of parameters rarely translates directly into a polytope in the space of coefficients of the characteristic polynomial. Therefore, to apply the result above it is often necessary to overbound the actual uncertainty set by a polytope, thereby introducing some conservatism into the test for stability.

In systems modeled by a linear state variable description, it is more often true (though hardly universal) that the set of uncertain parameters in the system can be mapped into a set of system matrices described by polytopes in the element space[1]. Given this motivation, it is natural to investigate generalizations of the edge result to the matrix case. Counter-examples have already appeared in the literature showing that stability of one dimensional edges is not sufficient to guarantee stability of a polytope of matrices[2]. The next logical generalization is to determine if stability of the polytope can be ascertained at all by tests of lower dimensional affine subsets, and if so, what dimension these sets would have. This paper shows that in many cases, stability of a polytope of matrices can be ascertained by examining 2n dimensional affine subsets of the boundary of the polytope.

II. REPRESENTATION OF MATRICES IN $\mathbb{R}^{nxn}$ AND $\mathbb{C}^{nxn}$

It will prove convenient to associate elements of $\mathbb{C}^{nxn}$ with those in $\mathbb{R}^{2n^2}$; to this end we define the operation vec[•]: $\mathbb{C}^{nxn} \rightarrow \mathbb{R}^{2n^2}$:

$$\mathbf{a} = \mathrm{vec}[A] :=$$

$$[\ \mathrm{Re}\{A_{11}\},\ \mathrm{Re}\{A_{12}\},\ \ldots,\ \mathrm{Re}\{A_{1n}\},\ \ldots,\ \mathrm{Re}\{A_{nn}\},\ \mathrm{Im}\{A_{11}\},\ \ldots,\ \mathrm{Im}\{A_{nn}\}\ ]^{T} \quad (2.1)$$

Note that to restrict our attention to matrices with real elements, we need only impose the constraint that

$$\left[\mathbf{O}_{n^2 x n^2} \ \ \mathbf{I}_{n^2 x n^2} \right] \mathbf{a} = \mathbf{0} \ . \quad (2.2)$$

III. REPRESENTATION OF POLYTOPES IN $\mathbb{R}^{nxn}$

Given the association of $\mathbb{C}^{nxn}$ with $\mathbb{R}^{2n^2}$ as indicated above, we may represent a polytope $\mathcal{P}$ as a set of equality constraints that determine the affine subspace containing the polytope (if $\mathcal{P}$ is not of full n^2 dimension) along with a set of inequality constraints that determine the hyperplanes which bound the polytope. Since our interest is restricted to polytopes of real matrices, we will also impose the equality constraints (2.2). Using superscript "IN" to denote quantities associated with inequalities, and superscript "EQ" to denote those associated with equalities, we have $\mathbf{a} \in \mathcal{P}$ if and only if $\mathbf{a}$ satisfies

---

[1] A simple example may be obtained by considering a linearization of the power system swing equations in which one or more transmission line susceptances lie in an interval.

$$\begin{bmatrix} \mathbf{M}^{IN} & \mathbf{O} \end{bmatrix} \mathbf{a} + \mathbf{d}^{IN} \le \mathbf{0} \tag{3.1}$$

and

$$\begin{bmatrix} \mathbf{M}^{EQ} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{n^2 \times n^2} \end{bmatrix} \mathbf{a} + \mathbf{d}^{EQ} = \mathbf{0}\,. \tag{3.2}$$

We will assume $\mathbf{M}^{IN}$ to have k rows, and $\mathbf{M}^{EQ}$ to have ℓ rows; note that it is necessary that $k + \ell$ exceed n^2 for $\mathcal{P}$ to be well defined.

IV. MAIN RESULT

THEOREM

Given a set of matrices whose elements lies in a polytope $\mathcal{P}$ as described above, with the property that $\ell < (n^2 - 2n)$. Suppose that for some $\mathbf{a} \in \mathcal{P}$, the associated real matrix **A** has an eigenvalue λ, with rank$[\lambda\mathbf{I} - \mathbf{A}] = n-1$ (i.e., the eigenspace associated with λ is of dimension 1). Then it follows that there exists an element **b** on the boundary of $\mathcal{P}$ with the property that the associated real matrix **B** has the same eigenvalue λ within its spectrum. Furthermore, **b** lies at a point where at least $n^2 - 2n - \ell$ of the inequality constraints defined by (3.1) are active; i.e., there exists an index set I, with elements selected from $\{1, 2, ..., k\}$, having $n^2 - 2n - \ell$ elements such that

$$\text{row}_i \left[\mathbf{M}^{IN} \ \ \mathbf{O} \right] \mathbf{b} + \mathbf{d}_i = \mathbf{0} \quad \text{for all } i \in I\,. \tag{4.1}$$

PROOF

Consider the set $\tilde{\mathcal{S}}$ defined as

$$\tilde{\mathcal{S}} = \{ \mathbf{C} \in \mathbb{C}^{n\mathrm{x}n} \mid \mathbf{C} = (\lambda\mathbf{I} - \mathbf{A})\mathbf{B}, \mathbf{B} \in \mathbb{C}^{n\mathrm{x}n} \}\,.$$

Note that $\tilde{\mathcal{S}}$ can be represented in $\mathbb{C}^{n^2}$ as the range of the matrix $n^2 \times n^2$ complex matrix diag$[(\lambda\mathbf{I}-\mathbf{A}),...,(\lambda\mathbf{I}-\mathbf{A})]$. By assumption, each diagonal block $(\lambda\mathbf{I}-\mathbf{A})$ has rank (in $\mathbb{C}^n$) of n-1. Therefore the $n^2 \times n^2$ block diagonal matrix will have rank n^2-n in $\mathbb{C}^{n^2}$, and its range can be described as the null space of an appropriate n x n^2 complex matrix $\tilde{\mathbf{N}}$; i.e., as the set $\{\mathbf{c} \in \mathbb{C}^{n^2} | \ \tilde{\mathbf{N}}\mathbf{c} = \mathbf{O}\}$. Finally, we note that this set must have an equivalent representation in $\mathbb{R}^{2n^2}$ as the null space of a 2n x $2n^2$ real matrix **N.** For this representation in $\mathbb{R}^{2n^2}$ we will drop the $\sim$ notation, and simply define

$$\mathcal{S} = \{\ \mathbf{c} \in \mathbb{R}^{2n^2} \ \big| \ \mathbf{Nc} = \mathbf{0}\}\ .$$

Note that by construction the set $\mathcal{S}$ is a linear subspace of $\mathbb{R}^{2n^2}$, and all complex matrices corresponding to points in $\mathcal{S}$ are singular, being rank deficient by at least 1.

The next step in the proof will be to construct from $\mathcal{S}$ an affine set whose elements represent complex matrices that share the eigenvalue λ. To this end, we define

$$\mathcal{S}_\lambda \ = \ \{\mathbf{b} = -\mathbf{c} + \mathrm{vec}[\lambda\mathbf{I}] \ \big| \ \mathbf{c} \in \mathcal{S}\ \} \ = \{\ \mathbf{b} \ \big| \ \mathbf{N}\,(\mathbf{b} - \mathrm{vec}[\lambda\mathbf{I}]\,) = \mathbf{0}\ \}\ . \tag{4.2}$$

Clearly, any complex matrix **B** corresponding to a point $\mathbf{b} \in \mathcal{S}_\lambda$ satisfies det $(\lambda\mathbf{I}\text{-}\mathbf{B}) = 0$; furthermore, this set contains the point **a** representing the original matrix **A**.

The final step of the proof constructs of the index set I by examining the intersection of $\mathcal{S}_\lambda$ with the polytope $\mathcal{P}$ and its boundary. Let I_0 initially equal the null set. Observe that any point in the intersection must simultaneously satisfy (3.2) and (4.2). By assumption, the number of constraints imposed by (3.2) and (4.2) is less than $2n^2$; together they define an affine subspace of $\mathbb{R}^{2n^2}$, of dimension one or greater, which contains the point $\mathbf{a} \in \mathcal{P}$. Being unbounded, this subspace must intersect one of the boundaries of $\mathcal{P}$ as defined by the rows of (3.1); denote index of the active row as i_1, and let $\mathbf{I}_1 = i_1$. This intersection defines a new affine subspace, having dimension decreased by at most one (if the new constraint is linearly independent from the proceeding). If the number of linearly independent constraints is $2n^2$, stop; the affine subspace so produced is a single point. Otherwise, the resulting set is still an affine subspace of dimension one or greater, and again must intersect a boundary of $\mathcal{P}$ defined by one of the remaining rows of (3.1). Denote the index of this active row i_2, and let $I_2 = I_1 \cup i_2$. This process is repeated until the number of linearly independent constraints on the real parts of elements of **a** reaches $2n^2$, and the resulting intersection is reduced to a single point. Note that this iteration can be repeated at least n^2 -2n - ℓ times, and hence the final index set will possess at least this number of elements.

□

COMMENTS

i) Observe that if one matrix in $\mathcal{P}$ possesses an given eigenvalue, the theorem above implies that there is a matrix contained in the intersection of the boundary of $\mathcal{P}$ with an affine subspace of $\mathbb{R}^{2n^2}$ that shares the same eigenvalue. In particular, if the polytope contains a matrix having an eigenvalue with non-negative real part, a matrix having an unstable eigenvalue (indeed, the same eigenvalue) must appear in this intersection. In the case when the constraints describing the subspace of matrices sharing this eigenvalue are independent of the constraints describing $\mathcal{P}$, this intersection will occur at point where $n^2 - 2n - \ell$ of the inequality constraints are active. Because the matrix **N** is not calculated (the theorem only shows its existance), we are left with $2n^2 - 2n$ known constraints active at the intersection; i.e. **b** must lie in a 2n dimensional subset of the boundary of $\mathcal{P}$.

ii) Recall that the Theorem assumes rank[λ**I** - **A**] = n-1. To relax this assumption and generalize the proof, one need only modify the construction of **N** in an obvious manner.

V. CONCLUSIONS

We have shown that for any polytope $\mathcal{P}$ of real matrices described by (3.1) and (3.2), associated with a matrix **A** having eigenvalue λ inside $\mathcal{P}$ is an affine set of matrices sharing this eigenvalue. Using arguments based on the dimension of this set, we have shown that its intersection with the boundary of $\mathcal{P}$ will contain a 2n dimensional affine set. While this set cannot be uniquely specified from given data, it must be one of a class of affine sets that are formed by forcing a specified number of the inequality constraints defined by (4.1) to be active. Perhaps the most important consequence of this result relates to testing if a given polytope $\mathcal{P}$ contains any matrices having unstable eigenvalues. One concludes that it is not necessary to test all matrices in $\mathcal{P}$, but only those lying in affine subspaces of the form described in the theorem above. While the theorem does not identify which of the inequality constraints of (3.1) will be active, it does indicate that the subspaces to be tested will have dimension no greater than 2n. An interesting question for further research would be determination of the lowest dimensional affine subspaces in the boundary having this property.

REFERENCES

[1] A. C. Bartlett, C. V. Hollot, and H. Lin, "Root locations of an entire polytope of polynomials: It suffices to check the edges," *Proc. 1987 American Control Conference*, pp. 1611-1616, Minneapolis, MN, June 1987.

[2] Fu, M. and Barmish, B. R., "Stability of Convex and Linear Combinations of Polynomials and Matrices Arising in Robustness Problems," *Proceedings of the Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore, 1987.

SLIDING REGIMES IN ANALOG SIGNAL ENCODING AND DELTA MODULATION CIRCUITS

HEBERTT SIRA-RAMIREZ,

Control Systems Department, Universidad de Los Andes, Mérida-Venezuela.

ABSTRACT

In this article the theory of Variable Structure Systems (VSS), and their associated Sliding Regimes, is proposed as a means of dealing with the problem of analog signal tracking in Delta Modulation Circuits.

1. INTRODUCTION

Sliding mode control of systems on discontinuity surfaces of the state space has been the subject of sustained research during the last 30 years, mainly by Soviet and East European researchers (Utkin [1],[2]). Recently, the technique has found renewed interest in the United States thanks to outstanding advances in microprocessor control applications and fast switching technologies with improved power ranges. For a thorough survey about the theoretical and practical achievements of the discipline the reader is referred to Utkin [2].

Section 2 of this article deals with a reapproachment to analog signal encoding in delta modulation circuits (Steele [3]) via Variable Structure Systems (VSS) theory and sliding modes. On-line analog signal tracking is viewed as a problem of inducing a sliding regime on a time-varying discontinuity surface defined in the extended phase space. Such surface is directly defined in terms of the codification error signal. The existence conditions for a sliding regime and the corresponding ideal (average) sliding dynamics naturally rederive the overload conditions, the overload frequency-dependent characteristics, the idling behavior and step responses in a number of practical encoder arrangements including Linear Delta Modulation (LDM), Delta-Sigma Modulation (DSM), Exponential Delta-Sigma modulation (EDSM) and Sylabically Companded modulation schemes.

Section 3 contains the conclusions and suggestions for further research in these areas. Basic results about VSS, and their associated sliding regimes defined on nonlinear nonstationary surfaces, are presented in the Appendix. The case is treated through a simple state augmentation procedure. General formulas are derived, within the context of nonlinear time-varying systems, for sliding mode existence conditions and ideal average sliding dynamics, both in the sense of Utkin [1] and of Filippov [4]

2. SLIDING REGIMES IN DELTA MODULATION CIRCUITS

2.1 Generalities

Analog signal tracking for encoding purposes is of primordial concern in every kind of encoding communication circuit. In particular, DM circuits, using local analog decoders in a feedback path, were proposed in a variety of nonadaptive and adaptive schemes just for the purpose of guaranteeing fast signal tracking while avoiding overload conditions and their associated encoding noise. Linear and adaptive DM schemes have been developed over the years since the end of World War II. Modern electronics has brought DM techniques into a full fledged digital era with the benefit of easily accesible digital circuit components and reliable algorithms of simpler and more efficient nature than their counterparts in analog

DM. However, a body of general theory was lacking in the existing literature. (See [3],[6]-[9]).

In this Section we propose to view delta modulation from the perspective of sliding regimes [1] induced on discontinuity surfaces defined in the extended state space of DM circuits which are described by differential equations with discontinuous right hand sides. The outcome of this program is a rederivation, in a more general setting, of a number of results which existed in the literature and were derived from purely heuristic grounds.

Only the classical LDM and SCDSM cases will be presented here. Digital encoding of signals will not be considered here. Digital DM, from the viewpoint of VSS, requires the development of a theory of sliding mode control for discrete-time systems.

2.2 An Ideal Variable Structure Systems Model for Linear Delta Modulation

The basic scheme of LDM encoding and decoding circuits is shown in Figure 2.1. In this circuit, the signal to be encoded for transmission, x(t), is continuously compared with the state of the **local decoder** (integrator) to generate an error signal e(t). The quantizer circuit generates a piecewise constant signal which is sampled during a small sampling interval τ with sampling period T. The resulting signal L(t) is integrated and filtered to recover x(t). The dynamical equations specifying the behavior of the modultation process are :

$$e(t) = x(t) - y(t) = x(t) - \int L(t)dt \tag{2.1a}$$

$$L(t) = \begin{cases} V \,\mathrm{sign}\, e(t) & \text{for } kT \le t < kT + \tau \\ 0 & \text{for } kT + \tau \le t < (k+1)T \end{cases} \tag{2.1b}$$

The transmitted signal L(t) is thus a sequence of regularly spaced pulses of duration τ, frequency 1/T, and amplitude of ± V according to the sign of the error signal at the moment of sampling. The signal y(t) is thus constituted by the sum of step functions of varying polarity, regularly delayed over the real line, which approximate the encoding signal x(t) within a quantization (granularity) step of value $\pm V\tau =: \pm \gamma \ (= \Delta x)$.

We may recast the tracking problem associated with LDM in terms of a sliding mode creation problem under certain high frequency sampling idealizations. The binary transmission aspects will be hence overlooked and deemed as independent of the tracking aspects. For this it is assumed that a suitable technology is available to transduce the high frequency outputs of the idealized sliding-mode tracking process into coded data suitable for transmission and subsequent decoding.

Assumptions 2.1: The basic idealizations needed for a Variable Structure Control (VSC) model of the modulation process are summarized as follows :

1) Pulse duration τ of sampling process is infinitely small
2) Sampling Frequency $f := 1/T$ is infinitely large
3) $\lim_{T \to 0,\ \gamma \to 0} \gamma/T = \lim_{f \to \infty,\ \gamma \to 0} \gamma f = V$

Using these assumptions the basic ideal VSC model, in extended state space

(y,θ), for LDM is simply :

$$e(\theta) = x(\theta) - y(\theta) \quad , \quad dy/dt = VL(\theta) = V \text{ sign } e(\theta) \; , \; d\theta/dt = 1 \qquad (2.2)$$

which can be reinterpreted in terms of the controlled evolution of the local decoder circuit state, col $[y,\theta]$, governed by :

$$dy/dt = Vu, \quad d\theta/dt = 1 \qquad (2.3)$$

geared to achieve a sliding motion on the following time-varying sliding surface defined in the extended phase space (y,θ),

$$S = \{ \; (y,\theta) \; R^2 : \; s(y,\theta) = e(\theta) = x(\theta) - y = 0 \; \} \qquad (2.4)$$

by action of the Variable Structure control law :

$$u = L(\theta) = \left\{ \begin{array}{ll} 1 & \text{for } s > 0 \\ -1 & \text{for } s < 0 \end{array} \right. = \text{ sign } s = \text{ sign } e \qquad (2.5)$$

Using the sliding mode existence conditions, given by (A.4), leads us directly to a well known result of LDM encoding.

A sliding regime exists on S if and only if :

$$\begin{array}{l} \lim_{s\to +0} ds/dt = \lim_{s\to +0} (\; dx/d\theta - V \;) \; < 0 \\ \lim_{s\to -0} ds/dt = \lim_{s\to -0} (\; dx/d\theta + V \;) \; > 0 \end{array} \qquad (2.6)$$

i.e, for all t :

$$-V < \min_\theta dx/d\theta \le dx/d\theta \le \max_\theta dx/d\theta < + V \qquad (2.7)$$

whose violation is the well known **slope overload condition** [3].

Under ideal sliding conditions $s= 0$, $ds/dt = 0$ one obtains :

$$x(t) = y(t) \quad ; \quad dx/dt = dy/dt \qquad (2.8)$$

i.e., ideal sliding is equivalent to perfect tracking of the encoding signal.

Summarizing, the basic consequences are the following :

1) Signal tracking is possible if and only if the slope overload condition is violated.
2) For x(t) identically zero the existence conditions are trivially satisfied and the tracking of "silence" is always accomplished in a chattering motion known as the **idling response**. If $x(t) = K =$ *constant*, the existence conditions are again trivially satisfied and a sliding motion is reached, from rest conditions, in finite time $t = V/K$. After this, the modulator exhibits again the idling response.
3) A step input violates the existence conditions (slope overload is trivially verified) in the time interval before the idling behavior starts.
4) For sinusoidal inputs $x(t) = E_s \sin 2\pi f_s t$, the existence conditions (2.6) translates into : $2\pi f_s E_s < V \; (\equiv \gamma f)$. This is the well known **overload frequency characteristic** for sinusoidal inputs in LDM (See[3] pp. 119).

5) The sliding mode existence conditions demand knowledge of the time derivative of the encoding input signal which can only be known on line. This clearly points the need of an adaptive scheme which guarantees the use of an appropiate on-line time-varying quantization limits $V(t)$, $-V(t)$ computed on the basis of the input signal $x(t)$.

Example 2.2 Figure 2.2 shows a computer simulated LDM response to a sinusoidal input of frequency 2.25 KHz, amplitude $E_s = 0.1$ mV and quatization limit $V = 2$ Volts. The initial state of the local decoder was set to -0.1 mV.

2.3 Sliding Mode Aspects of Delta-Sigma Modulation (DSM)

Classical DSM arose naturally from the impossibility of on-line accurate measurement of the time derivative of the encoding signal (this was easier to accomplish with Delayed Linear Delta Modulation). DSM proposes to encode and transmit the time integral of the input signal. In this manner, the overload conditions are given in terms of the original signal which is indeed on-line measurable. A substantial savings in hardware is accomplished on the remote decoder. The basic circuit is shown in Figure 2.3.

The model to be considered in this case, is :

$$de/dt = x(\theta) - Vu \ , \ d\theta/dt = 1 \ , \ u = \text{sign}\ e(\theta) \qquad (2.9)$$

The sliding surface is obtained from the desire of obtaining a zero encoding error $e(t)$:

$$S(e) = \{ \ s = e = 0 \ \} \qquad (2.10)$$

The ideal sliding conditions $s = 0$, $ds/dt = 0$ lead to :

$$e = 0 \ , \ u_{EQ}(\theta) = x(\theta)/V \qquad (2.11)$$

The existence conditions (A.9) for a sliding mode on $S(e)$ translate into :

$$-V < \min_{\theta} x(\theta) < x(\theta) < \max_{\theta} x(\theta) < V \qquad (2.12)$$

or briefly, for all t , $|x(t)| < V$ i.e., the slope overload condition is transformed into an **amplitude overload condition**. In particular, for sinusoidal inputs, $x(t) = E_s \sin 2\pi f_s t$, the overload condition is frequency independent and of the form $E_s < V \ (\equiv \gamma f \ )$ (See [3] pp. 18) .

The basic consequence of a sliding mode approach to DSM is the rederivation of the folowing known facts :

1) Idling behavior of DSM, corresponding to zero input response, is due to trivial satisfaction of of sliding mode existence conditions (amplitude overload violation). Step response exhibits idle behavior if and only if step amplitude is within quantizer limit bounds. The idling behavior is the immediate reaction of the modulator to a step input.

2) Overload characteristics are independent of the frequency of the sinusoidal input and hence the DSM coding-decoding process is suitable for a wide range of applications.

2.4 Sliding Modes in Exponential Delta-Sigma Modulation (EDSM)

EDSM is quite similar to DSM except for the use of a "leaky" integrator which provides additional local feedback for stabilization purposes of the error signal. The basic circuit is shown in Figure 2.4. The variable structure model of this circuit is :

$$de/dt = -(1/RC)e + (1/RC)[x(\theta) - u\bar{V}] , \quad d\theta/dt = 1, \quad u = \text{sign } e(\theta) \tag{2.13}$$

The sliding surface is represented by

$$S(e) = \{ e : s = e = 0 \} \tag{2.14}$$

As before, the ideal sliding conditions are obtained by setting $s = 0$ and $ds/dt = 0$ i.e.,

$$e = 0 \quad ; \quad u_{EQ}(\theta) = x(\theta)/\bar{V} \tag{2.15}$$

The existence conditions (A.9) again translate into violation of an **amplitude overload condition** of the form $|x(\theta)| < \bar{V}$.

2.5 Sylabically Companded Delta-Sigma Modulation (SCDSM)

SCDSM constitutes an adaptive scheme implemented on a SDM circuit. The local demodulator is complemented with a nonlinear circuit which has a decoder for the transmitted binary signal. The local demodulator rectifies the decoded signal, takes the envelope through an envelope detector and rises the resulting signal to a certain integer power. A constant positive polarization signal is subsequently added and the resulting signal is multiplied with the binary coded error signal. The resulting signal is compared with the analog input signal and a slope error is formed. Afer leaky integration, the error signal is generated. Figure 2.5, ([3] pp. 185), shows the basic scheme of the adaptive arrangement at the encoding portion.

It is quite simple to show that the local decoder of the SCDSM simply realizes the variable structure control law (A.13) and hence a local sliding regime is guaranteed to exist on the time varying sliding surface $S(x,\theta)$ of the ESDM scheme as given by (2.14).

3. CONCLUSIONS

Sliding regimes have been shown to be relevant in the analysis of analog signal tracking devices such as those performing encoding for signal transmission over digital communication channels.

The necessary and sufficient conditions for the existence of a sliding regime determine the capabilities of the dynamic tracking arrangement. In Delta Modulation systems, the existence conditions naturally point to the basic need of an adaptive modulation scheme. The slope overload condition is shown to be a direct consequence of the well known sliding regime existence conditions. The Sylabically Companded schemes are naturally rederived using an equivalent-control-based variable structure controller. An extension of the results obtained in this paper to modern digital delta modulation requires the development of a theory of sliding regimes for discrete-time dynamical systems. The method can also be used in phase modulation arrangements, geared towards betterment of power factors in inductive loads of AC lines. This is proposed as a topic for further work.

REFERENCES

[1] Utkin, V.I., " *Sliding Modes and their Application in Variable Structure Systems,* " MIR Publishers, Moscow 1978.

[2] Utkin, V.I., "Discontinuous Control Systems: State of the Art in Theory and Applications," 10th World Congress on Automatic Control. Munich, July 1987 (preprint).

[3] Steele, R., *Delta Modulation Systems,* Pentech Press Ltd., London 1975.

[4] Filippov, A., " A Differential Equation with a Discontinuous Right-Hand Side, " *Matematicheskii sbornik,* Vol. 51, No. 2, 1960.

[5] Sira-Ramirez, H., "Differential Geometric Methods in Variable Structure Control," *International Journal of Control.* (Submitted for Publication)

[6] C.L. Song, J. Garodnick and D.L. Schilling, " A Variable-Step-Size Robust Delta Modulator, " *IEEE Transactions on Communication Technology.* Vol. COM-19, No. 6 . pp. 1033-1044. December 1971.

[7] L.H. Zetterberg and J. Uddenfeld, " Adaptive Delta Modulation with Delayed Decision," *IEEE Transactions on Communications,* Vol. COM-22, No. 5, pp. 1195-1198, September 1974.

[8] V.R. Dhadesugoor, C. Ziegler and D.L. Schilling, " Delta Modulators in Packet Voice Networks, " *IEEE Transactions on Communications,* Vol. COM-28, No. 1, pp. 33-51, January 1980.

[9] J. Dunham, " Optimal Discrete-Time Delta Modulation System, " *IEEE Transactions on Communications,* Vol. COM-34, No. 5, pp 510-512, May 1986.

[10] H.Sira-Ramirez, " Variable Structure Control of Nonlinear Systems," *International Journal of Systems Science,* (accepted for publication, to appear).

Acknowledgments The author is indebted to Professors Román A. Contreras and Ermanno Pietrosemoli, of the Electrical Engineering Department, for useful criticism and for providing many of the references used in the preparation of this article. This work was supported by the CDCHT of the Universidad de Los Andes under research contract No. I-280-87.

APPENDIX

Sliding Regimes on Nonstationary Discontinuity Surfaces

In this section we shall present some of the basic aspects of sliding modes of nonlinear dynamical systems on time-varying sliding surfaces. The reader is referred to Sira-Ramirez [5,10] for an exposition on the stationary sliding surface case.

Consider the nonlinear time-varying dynamical system defined on an open set X x T of R^n x R and expressed in local coordinates of the extended state space (x,θ) as :

$$dx/dt = f(x,\theta) + u\, g(x,\theta) \qquad d\theta/dt = 1 \tag{A.1}$$

with f and g being locally **defined time-varying smooth vector fields** of R^n. (*col* [f, 1] and *col* [g,0] are also called **direction vector fields**). The control function u is assumed to be of discontinuous nature, taking values on the discrete set U = { -1,1 } , according to the switching logic :

$$u = \text{sign}\ s(x,\theta) \tag{A.2}$$

where $s(x,\theta)$ is a smooth function, of time varying nature, with uniformly nonzero gradient on X x T. $s : R^n \times R \to R$, which defines a discontinuity surface, in the extended state space, of the form :

$$S(x,\theta) = \{ (x,\theta) \in R^{n+1} : s(x,\theta) = 0 \} \tag{A.3}$$

Definition A.1 [1] A local **Sliding Regime** is said to exist on $S(x,\theta)$ if and only if, locally on X x T, the controlled motion (A.1),(A.2) is such that :

$$\lim_{s \to +0} ds/dt < 0 \quad \text{and} \quad \lim_{s \to -0} ds/dt > 0 \tag{A.4}$$

The trajectories converge, locally and uniformly in T, towards the discontinuity surface $S(x,\theta)$ where they locally undergo a chattering motion constraining them, in the average, to $S(x,\theta)$. As a consequence of this, the sliding motions ideally adopt $S(x,\theta)$ as a **local integral manifold**. Choosing an appropriate discontinuity surface, the system dynamics is changed at will with charcteristics usually not present in any of the intervening controlled structures represented by (A.1),(A.2) on each side of $S(x,\theta)$.

By a **local property**, it will be henceforth understood that it holds true locally on X and uniformly on the open interval T. It is assumed that X has a nonempty intersection with $S(x,\theta)$ for all θ in T.

Lemma A.2 [5] If a sliding motion exists on $S(x,\theta)$ then the following transversality condition is locally verified on $S(x,\theta)$:

$$[\partial s/\partial x]^T g(x,\theta) < 0 \tag{A.5}$$

The solutions of (A.1),(A.2) are locally defined everywhere on X x T, except on the surface $S(x,\theta)$ where the right hand side of (A.1) undergoes discontinuities. Basically, two kinds of solutions have been proposed to describe the resulting motion on $S(x,\theta)$, in an ideal manner which neglects switch inertias and imperfections due to perturbations and modelling errors. The first one is due to Filippov [4], and the second is due to Utkin [1] . For the linear in the control case, such as (A.1), both approaches are totally equivalent.

Ideally, on $S(x,\theta)$, the motions of the controlled system can be described as if controlled by a smooth feedback control function known as the equivalent control. Such a function is defined from the following manifold invariance conditions :

$$ds/dt\,|_{s=0} = 0 \tag{A.6}$$

where the total time derivative of $s(x,\theta)$ is computed along local solutions of (A.1) and satisfied for certain smooth control function known as the **equivalent control**, here denoted by $u_{EQ}(x,\theta)$, i.e.,

$$[\partial s/\partial x]^T[\ f(x,\theta) + u_{EQ}(x,\theta)\ g(x,\theta)\ ] + \partial s/\partial\theta \ = 0 \ ,$$

or explicitly :

$$u_{EQ}(x,\theta) = -\ ([\partial s/\partial x]^T g(x,\theta))^{-1}[\partial s/\partial x]^T\ f(x,\theta) - ([\partial s/\partial x]^T g(x,\theta)\ )^{-1}[\partial s/\partial\theta] \qquad (A.7)$$

From (A.7) and Lemma A.2 it easily follows that a necessary condition for the existence of a local sliding motion on $S(x,\theta)$ is that $u_{EQ}(x,\theta)$ locally exists. It is also straightforward to show that, ruling out the trivial case $[\partial s/\partial x]^T f(x,\theta) = 0$ and $[\partial s/\partial x]^T g(x,\theta) = 0$ whereby for any control function S is invariant, the equivalent control, if it exists, is also unique ([5]).

The ideal sliding dynamics is obtained from using (A.7) on (A.1) with the formal substitution of u by $u_{EQ}(x,\theta)$. This is the basis of the **Equivalent Control Mehtod** [1]:

$$dx/dt \ = \ \{I-g(x,\theta)[(\partial s/\partial x)^T g(x,\theta)]^{-1}(\partial s/\partial x)^T\}f(x,\theta)-g(x,\theta)[(\partial s/\partial x)^T g(x,\theta)]^{-1}(\partial s/\partial\theta) \qquad (A.8)$$

The vector field defining the ideal sliding dynamics belongs to the time-varying tangent space at x of $S(x,\theta)$.

Theorem A.3 A sliding motion locally exists on $S(x,\theta)$ if and only if :

$$-1 < u_{EQ}(x,\theta) < 1 \qquad (A.9)$$

Proof: (See [5])

Finally, we present prescriptions for a family of variable structure control laws which locally guarantee the existence of a sliding motion on the discontinuity surface $S(x,\theta)$, provided $S(x,\theta)$ qualifies as a sliding surface. The switching laws are applicable to those systems where a feedback control set of the form $\{u^+(x,\theta),\ u^-(x,\theta)\ \}$ is sought which locally generates a sliding motion, rather than having one such set already fixed in the form $U = \{\ -1,1\ \}$. The control functions $u^+(x,\theta)$, $u^-(x,\theta)$ are required to be smooth time-varying feedback control laws with $u^+(x,\theta) < u^-(x,\theta)$ locally in $X \times T$. For the class of systems under consideration, the existence condition (A.4) adopts the form [1]:

$$u^+(x,\theta) \ < \ u_{EQ}(x,\theta) \ < u^-(x,\theta) \qquad (A.10)$$

Proposition A.4 If a local sliding regime exists on $S(x,\theta)$ and $u_{EQ}(x,\theta)$ is not identically zero on an open set of $X \times T$, then there exists a constant $K > 1$ such that the switching logic $u = -\ K\ |u_{EQ}(x,\theta)|\ \mathrm{sign}\ s(x,\theta)$ creates a local sliding regime on $S(x,\theta)$.

Proof Obvious

Since it is entirely possible that $u_{EQ}(x,\theta) \equiv 0$, the following prescriptions, of importance in adaptive delta modulation cases, guard against this possibility. The proofs of these propositions are left to the reader.

Proposition A.5 If a local sliding regime exists on $S(x,\theta)$ then there exists constants $K > 1$ and $A > 0$ such that the switching logic :

$$u = \left(A + K \,|u_{EQ}(x,\theta)|\right) \operatorname{sign} s(x,\theta) \qquad \text{(A.11)}$$

creates a local sliding regime on $S(x,\theta)$.

Finally, let env $z(x(t),t)$ denote the function specified by :

$$\mathit{env}\, z(x(t),t) = \max \{ \sup_{\tau \le t} | z(x(\tau),\tau)| \,,\, | z(x(t),t) | \} \qquad \text{(A.12)}$$

By $\mathit{env}^{\,n}\, z(x(t),t)$ is denoted the n-th power of the values of $\mathit{env}\, z(x(t),t)$

Proposition A.6 If a local sliding regime exists on $S(x,\theta)$ then there exists constants $K > 1$, $A > 0$ and an integer $n \ge 1$ such that the switching logic :

$$u = \left(A + K \;\mathit{env}^{\,n}\, |u_{EQ}(x,\theta)|\right) \operatorname{sign} s(x,\theta) \qquad \text{(A.13)}$$

creates a local sliding regime on $S(x,\theta)$.

FIGURES

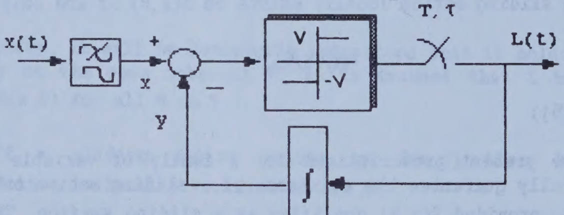


Figure 2.1 Linear Delta Modulation Encoder

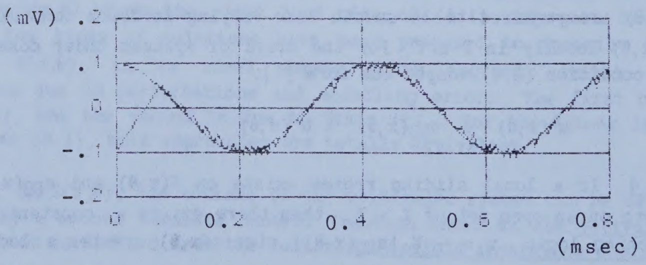


Figure 2.2 Typical Sinusoidal Response of LDM

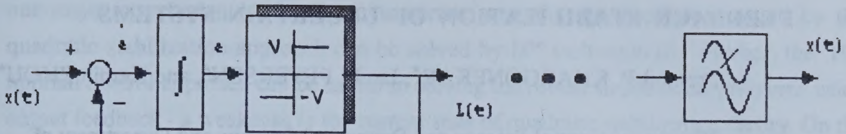

+
e
x(t)
−
e
V
-V
L(t)
x(t)

Figure 2.3 Delta-Sigma Modulator and Demodulator Circuits

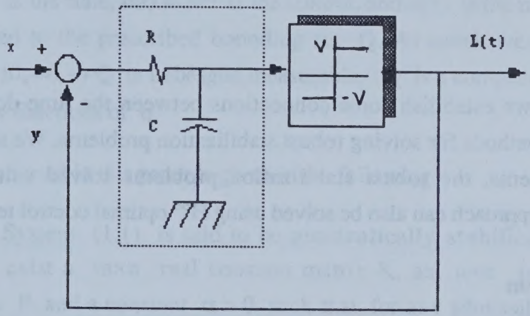


Figure 2.4 Exponential Delta-Sigma Modulation Circuit

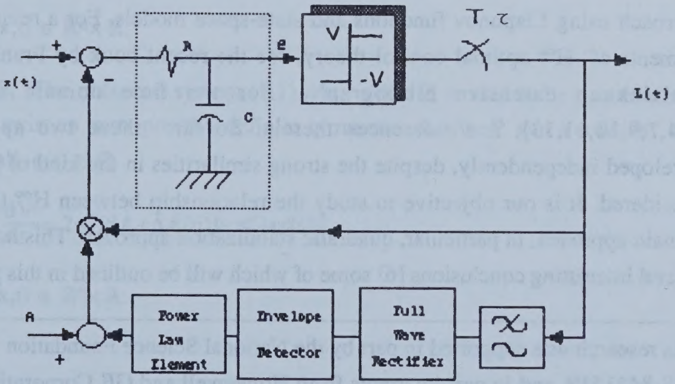

+
R
e
V
-V
T
τ
x(t)
−
L(t)
C
A
+
Power Law Element
Envelope Detector
Full Wave Rectifier

Figure 2.5 Sylabically Companded Delta-Sigma Modulation

FEEDBACK STABILIZATION OF UNCERTAIN SYSTEMS

Pramod P. KHARGONEKAR*, Ian R. PETERSEN#, and Kemin ZHOU*

*Center for Control Science and Dynamical Systems and Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455.

#Department of Electrical and Electronic Engineering, University of New South Wales, Australian Defence Force Academy, Campbell, ACT 2600, Australia.

Abstract

In this paper, we estabilish some connections between the time domain and the H^∞ optimization methods for solving robust stabilization problems. We show that for linear uncertain systems, the robust stabilization problems solved using the *quadratic stabilization* approach can also be solved using H^∞ optimal control techniques.

I. Introduction

Robust control theory deals with control of systems with modeling uncertainty and is among one of the most vibrant areas of research in control theory. In this paper, we will focus on two major research directions in the robust control theory: the frequency domain approach involving input-output models and H^∞ optimization tools and the time domain approach using Liapunov functions and state-space models. For a review of the basic elements of H^∞ optimal control theory, see the recent book by Francis, which also contains an extensive bibliography. For the time domain approach, see [1,4,7,9,10,11,13], and references therein. So far, these two approaches have developed independently, despite the strong similarities in the kind of problems being considered. It is our objective to study the relationship between H^∞ theory and time domain approach, in particular, quadratic stabilization approach. This has already led to several interesting conclusions [6] some of which will be outlined in this paper. Indeed,

---

This research was supported in part by the National Science Foundation under grant No. ECS-8451519, and in part by grants from Honeywell and GE Corporations, and in part by the Australian Research Grants Commission.

our major conclusion is: for linear uncertain systems, the problems solved by the quadratic stabilization approach can be solved by H^∞ technique [6]. Further, the H^∞ optimal control approach can be useful in solving the robust stabilization problems using output feedback - a weakness in the current state of quadratic stabilization theory. On the other hand, the quadratic stabilization approach leads to new results in H^∞ optimal control theory, especially in problems involving state-feedback [5, 6].

Consider the uncertain dynamical system described by differential equation

$$\frac{dx}{dt} = [A+\Delta A(q(t))]x + Bu \tag{1.1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control, and $q(t)$ is the model uncertainty which is restricted to the prescribed bounding set $\mathbb{Q}$. As usual, we assume that the uncertainty $q(.): [0, \infty) \to \mathbb{Q}$ is Lebesgue measurable, $\mathbb{Q}$ is a compact set, and ΔA is continuous matrix functions of q.

The stability notion used in this paper is given in the following

Definition 1.2. System (1.1) is said to be **quadratically stabilizable via linear control** if there exist a $m\times n$ real constant matrix K, an $n\times n$ positive definite symmetric matrix P and a constant $\alpha > 0$ such that, for any admissible uncertainties q(t), the closed-loop system with state feedback $u(t) = -Kx(t)$ and Lyapunov function $V(x) = x^TPx$ has the following property

$$L(x,t) := \frac{dV}{dt} = 2x^TP[A+\Delta A(q)]x - 2x^TPBKx \leq -\alpha\|x\|^2$$

for all pairs $(x,t) \in \mathbb{R}^n\times \mathbb{R}$.

Definition 1.3. The unforced system (1.1), i.e., $u(t) = 0$, is said to be **quadratically stable** if there exist an $n\times n$ positive definite symmetric matrix P and a constant $\alpha > 0$ such that, for all $q(t) \in \mathbb{Q}$

$$L(x,t) := \frac{dV}{dt} = 2x^TP[A+\Delta A(q)]x \leq -\alpha\|x\|^2$$

for all pairs $(x,t) \in \mathbb{R}^n\times \mathbb{R}$.

As is well known, if the above inequalities in definitions 1.2 and 1.3 hold, then the closed loop system is uniformly asymptotically stable at the equilibrium point $x = 0$, for any given admissible uncertainties $q(t)$.

II. Quadratic Stability and Small Gain Theorem

The following theorem is the key to this paper, it may be known, however, the present authors are not able to find any references.

Theorem 2.1.(See Khargonekar, Petersen and Zhou [6]) Consider a class of uncertain systems described as follows

$$\frac{dx}{dt} = \{A_o + DF(t)E\}x(t), \qquad t \geq 0$$

$$F(t) \in \mathbb{F} \qquad\qquad (\Sigma)$$

where $\mathbb{F} := \{F(t) : F^T(t)F(t) \leq \gamma^2 I$; The elements of $F(t)$ are Lebesgue measurable.$\}$. Then system (Σ) is quadratically stable if and only if it satisfies the following conditions:

(i) A_o is a stability matrix.

(ii) There exists a constant $\varepsilon > 0$ such that

$$\gamma \| E(sI-A_o)^{-1}D \|_\infty \leq 1- \varepsilon.$$

Remark 2.2. Theorem 2.1 states that "small gain theorem" is both necessary and sufficient for quadratic stability of system (Σ). This has significant implications, for example, see section V. We also note that this theorem is stronger than Theorem 2.1 and Theorem 2.2 of Petersen [8]

III. Robust Stabilization via Output Feedback

Consider the following dynamical system

$$\frac{dx}{dt} = [A + \Delta A]x(t) + Bu(t) , \qquad t \geq 0 \qquad (3.1a)$$

$$y = Cx \qquad (3.1b)$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control, and $y(t) \in \mathbb{R}^k$ is the output.

Assumptions 3.2. Throughout this section, it will be assumed that the system (3.1) satisfies the following assumptions:

A1. (A, B) is stabilizable and (A, C) is detectable ;
A2. $B \in \mathbb{R}^{n \times m}$ is full rank, $C \in \mathbb{R}^{k \times n}$ is full rank ;

For simplicity of exposition, let us also assume

A3. $\Delta A = BF(t)$ and $F(t) \in \mathbb{F}$.

The assumption A3 is not necessary, and the following results extend to much more general situations. But, the proof gets somewhat tedious. However, this special case is ample to illustrate our point: CLP based methods may be regarded as providing sufficient conditions to guarantee that the infimal cost of the associated H^∞ optimal control problem is *zero* as shown in Theorem (3.6). This has further consequences which are discussed in Remark (3.7).

The problem of stabilization of this *time-varying* uncertain system using output feedback is as yet unsolved in the uncertain systems robust stabilization literature. However, using Theorem 2.1, it is fairly clear that this problem can be completely solved using H^∞ optimization technique. Indeed, we have

Theorem 3.3. Let $H_o(s) := \{sI-A+BK(s)C\}^{-1}B$ and define

$$\varepsilon_o := \inf \{ \| H_o(s) \|_\infty : K(s) \in \mathbb{S}_o \}$$

where $\mathbb{S}_o := \{ K(s) : K(s)$ internally stabilizes the nominal system (3.1)$\}$. Then the robust stabilization problem is solvable iff

$$\varepsilon_o < 1/\gamma .$$

Now we only need to note that there exists a standard procedure to obtain ε_o and the stabilizing controller $K(s)$ such that $\| H_o(s) \|_\infty < 1/\gamma$.

Galimidi and Barmish (1986) have given a sufficient condition and a procedure to solve the robust stabilization problem in a special case. It is illuminating to examine their procedure from the H^∞ perspective. This is done next.

The most important result shown in Galimidi and Barmish (1986) is the following

Theorem 3.4. (See Galimidi and Barmish [3]). System (3.1) satisfying the assumptions A1-A3 is robustly stabilizable via a dynamic output feedback if there exist $P > 0$ and $\Omega \in \mathbb{R}^{m \times k}$ such that the following CLP is solved:

(i) $\Theta^T(A^TP + PA)\Theta < 0$

(ii) $B^TP = \Omega C$

where $\Theta \in \mathbb{R}^{n \times (n-k)}$ satisfies

$$C\Theta = 0 \text{ and } \Theta^T\Theta = I.$$

In the following, we study this problem from the H^∞ perspective. The following simple result is key to the main results of this section.

Lemma 3.5.(See Khargonekar, Petersen and Zhou [6]) Given (A, B, C) satisfying assumptions 3.2, suppose there exist $P > 0$ and Ω such that the CLP in Theorem 3.4 is solved. Then there exists an L such that $A - LC$ is a stability matrix and the transfer function matrix

$$T(s) := \Omega C(sI - A + LC)^{-1}B$$

is an outer function matrix.

The following theorem makes an explicit connection between the CLP based approach and H^∞ optimal control solution to the robust stabilization problem in the output feedback case.

Theorem 3.6.(See Khargonekar, Petersen and Zhou [6]) Let $H_1(s) := \{sI-A+BK(s)C\}^{-1}B$ and suppose there exist $P > 0$ and Ω such that the CLP in Theorem 3.4 is solved. Then

$$\varepsilon_1 := \inf \{ \| H_1(s) \|_\infty : K(s) \in \mathbb{S}_o \} = 0.$$

Remark 3.7. Theorem 3.3 and Theorem 3.6 together clearly imply that whenever the robust stabilization problem is solvable via CLP method, then it can be solved via H^∞ technique. It is obvious from Theorem 3.6 that CLP method can only solve a very small class of uncertain system robust stabilization problems. In fact, Theorem 3.6 implies that if CLP is solvable, then the system is robustly stabilizable given an a priori bound, no matter how large, on the norm of system uncertainty F(t).

IV. Stabilization (via State Feedback) Using Quadratic Overbounding

Consider the following uncertain dynamical system with structured uncertainties

$$\frac{dx}{dt} = [A + \sum_{i=0}^{p} D_i F_i(t) E_i]x + Bu , \qquad t \geq 0, \qquad (4.1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $D_i \in \mathbb{R}^{n \times li}$, $E_i \in \mathbb{R}^{ki \times n}$ are constant known matrices, and $F_i(t) \in \mathbb{F}_i$ are norm bounded time-varying Lebesgue measurable matrix functions representing modeling or parameter uncertainty, where $\mathbb{F}_i := \{F_i(t) \in \mathbb{R}^{li \times ki} : F_i^T(t)F_i(t) \leq \gamma^2 I$; The elements of $F_i(t)$ are Lebesgue measurable.$\}$. This model is a generalization of the system model considered in Section III and allows for more structure in the system uncertainty. Define block matrices

$$D := [\ D_1\ D_2\ ..\ D_p], \quad E := \begin{bmatrix} E_1 \\ E_2 \\ : \\ E_p \end{bmatrix}, \text{ and } \quad F_o(t) := \begin{bmatrix} F_1(t) & & \\ & F_2(t) & \\ & : & \\ & & F_p(t) \end{bmatrix}.$$

Then a very conservative approach is the following

Theorem 4.2. Let $H_2(s) := E\{sI-A+BK(s)\}^{-1}D$ and define

$\varepsilon_2 := \inf \{ \| H_2(s) \|_\infty : K(s) \in \mathbb{S} \}$

where $\mathbb{S} := \{ K(s) :$ K(s) internally stabilizes the nominal system (4.1)$\}$. Then the robust stabilization problem is solvable *if*

$\varepsilon_2 < 1/\gamma$.

Output feedback can also be considered very easily using Theorem 4.2. However, our purpose is only to compare the H^∞ approach and quadratic overbounding method. The following theorem due to Khargonekar, Petersen, and Rotea [5] makes clear that whenever the above problem is solvable by Theorem 4.2 via dynamic state feedback, then it is solvable via a constant state feedback.

Theorem 4.3. (See Khargonekar,Petersen,and Rotea [5])Let $H_s(s) := E(sI-A+BK)^{-1}D$ denote the transfer function with the *static* state feedback u = -Kx and let $H_d(s) := E\{sI-$

$A+BK(s)\}^{-1}D$ denote the same transfer function with the *dynamic* state feedback $u = -K(s)x$. Define

$$\varepsilon_s := \inf \{ \| H_s(s) \|_\infty \ : \ K \in \mathbb{R}^{m \times n} \text{ such that } A-BK \text{ is a stability matrix } \}$$
$$\varepsilon_d := \inf \{ \| H_d(s) \|_\infty : K(s) \in \mathbb{S} \}.$$

Then $\varepsilon_d = \varepsilon_s$.

A Riccati equation approach is also used by Petersen and Hollot [9] to obtain the stabilizing control law for this uncertain system; See also the extensions by Schmitendorf [10], Zhou and Khargonekar [13].

Let $T := \gamma DD^T = \sum_{i=0}^{p} \gamma D_i D_i^T$, $U = \gamma E^T E = \sum_{i=0}^{p} \gamma E_i^T E_i$.

Theorem 4.4.(See Petersen and Hollot [9]) The system (4.1) is quadratically stabilizable via state feedback if (for any given $Q > 0$) the following Riccati equation

$$A^TP + PA - \frac{1}{\varepsilon}PBB^TP + PTP + U + \varepsilon Q = 0 \tag{4.5}$$

has a positive definite solution P for some $\varepsilon > 0$. Further the feedback gain can be chosen as

$$K = \frac{1}{2\varepsilon}B^TP. \tag{4.6}$$

Now we shall show that if (4.5) and (4.6) is satisfied the Theorem 4.2 is satisfied.

Theorem 4.7. If A, B, E_i, and D_i satisfy the condition in Theorem 4.4 and K is as in (4.6). Then

$$\| E(sI - A + BK)^{-1}D \|_\infty < 1/\gamma.$$

Proof: The result follows by rewriting (4.5) as

$$(A - BK)^TP + P(A - BK) + \gamma PDD^TP + \gamma E^TE + \varepsilon Q = 0$$

and using again Lemma 1 of Willems [12]. []

Theorem 4.2, 4.3, 4.4, and 4.7 together clearly imply that whenever the problem is solvable via quadratic overbounding method, then it is solvable via H^∞ technique.

References

[1] B. R. Barmish, " Necessary and sufficient conditions for quadratic stabilizability of an uncertain linear systems, " Journal of Optimization Theory and Applications, Vol. 46, No.4, August 1985, pp. 399-408.

[2] B. A. Francis, *A Course in H^∞ Control Theory.* Springer-Verlag:Berlin, 1987.

[3] A. R. Galimidi and B. R. Barmish, " The constrained Lyapunov problem and its application to robust output feedback stabilization," IEEE Transactions on Automatic Control, Vol. AC-31, No. 5, May 1986, pp. 410-419.

[4] C. V. Hollot and B. R. Barmish, " Optimal quadratic stabilizability of uncertain linear systems," Proceeding of 18th Allerton Conference on Communication, Control and Computation, 1980, University of Illinois, Monticello.

[5] P. P. Khargonekar, I. R. Petersen, and M. A. Rotea, " H^∞-optimal control with state-feedback", Technical Report, Dept. of Electrical Engineering, August 1987.

[6] P. P. Khargonekar, I. R. Petersen, and K. Zhou,"Robust stabilization of uncertain systems and H^∞ optimal control", in preparation.

[7] G. Leitmann, " Guaranteed asymptotic stability for some linear systems with bounded uncertainties, " Journal of Dynamic Systems, Measurement, and Control, Vol. 101, September, 1979, pp. 212-216.

[8] I. R. Petersen, " Notions of stabilizability and controllability for a class of uncertain linear systems, " International Journal of Control, 1987, Vol. 46, No. 2, pp. 409-422.

[9] I. R. Petersen and C. V. Hollot, " A Riccati equation approach to the stabilization of uncertain linear systems, " Automatica, Vol. 22, No. 4, 1986, pp. 397-411.

[10] W. E. Schmitendorf, " Designing stabilizing controllers for uncertain systems using the Riccati equation approach," Proceedings of 1987 American Control Conference, Minneapolis, June 1987.

[11] J. S. Thorp and B. R. Barmish, " On guaranteed stability of uncertain linear systems via linear control, " Journal of Optimization Theory and Applications, Vol. 35, No. 4, December 1981, pp. 559-579.

[12] J. C. Willems, " Least squares stationary optimal control and the algebraic Riccati equation," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, December 1971, pp. 621-634.

[13] K. Zhou and P. P. Khargonekar, " Stabilization of uncertain systems via linear feedback control," submitted for publication, March 1987.

Observer-based Design of Uncertain Systems: Recovering State Feedback Robustness Under Matching Conditions

F. Esfandiari and H.K. Khalil

Department of Electrical Engineering
Michigan State University
East Lansing, MI 48824

Abstract- In this paper a high-gain observer is proposed such that the observer-based control recovers the stability robustness of certain linear and nonlinear state feedback controls.

1.Introduction

In the past two decades, various linear and nonlinear state feedback controllers have been proposed for the stabilization of linear systems in the presence of time-varying parametric uncertainties [1-3 to name a few], where a priori known nonstatistical information about the uncertainties are used in the control design. Recently there have been efforts to develop output feedback controllers based on such methods [4-6, among others].

Basically the problem is to design an observer in a way that the observer-based control recovers the robustness properties of the state feedback design. In [4] this problem is solved for the case when the relative degree of the system is one and the state feedback controller is linear. Some extentions of the ideas introduced in [4] can be found in [5]. In [6] a Riccati equation approach is used, but the stability of a nearby system rather than the original system has been proved.

In this paper we address the problem of observer-based control for minimum phase, left-invertible systems, where the relative degree can be higher than one. We also consider a large class of state feedback designs, which includes the nonlinear controllers of [1], as well as the linear controllers of [2-6]. A special coordinate basis [7] and singular perturbation theory are used to perform the analysis.

2.Problem Statement

Consider the linear time-invariant system (2.1)

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + E\delta(x,u,t) \\ y(t) = Cx(t) \end{cases} \tag{2.1}$$

where $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, $y \in \mathbf{R}^p$ and $\delta(.) \in \mathbf{R}^l$.

Assumption 1: The disturbance vector, $\delta(.)$, satisfies the following inequalities.

$$\| \delta(x,u,t) \| \leq \delta_x \|x\| + \delta_u \|u\| + \delta_c$$

$$\| \delta(x,u,t) - \delta(x,\hat{u},t) \| \leq k_1 \| u - \hat{u} \|$$

where δ_x , δ_u , δ_c and k_1 are nonnegative constants and δ_u is strictly less than one.

Assumption 2: There exist a state feedback control $u = -f(x)$ and a Lyapunov function $W(x)$ for which the following inequalities hold.

$$\frac{\partial W}{\partial x}\Big[Ax - Bf(x) + E\delta(x, -f(x), t)\Big] \le -\beta_4 \| x \|^2 + (\delta_c \beta_3 + \beta_2) \| x \| + \delta_c \beta_1$$

$$\underline{\eta}_w \| x \|^2 \le W(x) \le \overline{\eta}_w \| x \|^2$$

$$\Big\| \frac{\partial W}{\partial x} \Big\| \le k_2 \| x \|$$

$$\| f(x) - f(\hat{x}) \| \le k_3 \| x - \hat{x} \|$$

$$\| f(x) \| \le k_4 \| x \| + k_5$$

where β_4, $\underline{\eta}_w$, $\overline{\eta}_w$, k_2, k_3 are positive constants, while β_3, β_2, β_1, k_4 and k_5 are nonnegative constants.

Remarks

i) The first inequality ensures uniform ultimate boundedness of the closed-loop system, with respect to a set Ω, where

$$\Omega \triangleq \left\{ x \in \mathbf{R}^n \mid W(x) < \overline{\eta}_w c_*^2 \right\} \tag{2.2}$$

$$c_* \triangleq \frac{\delta_c \beta_3 + \beta_2}{2\beta_4} + \left[\frac{(\delta_c \beta_3 + \beta_2)^2}{4\beta_4^2} + \frac{\delta_c \beta_1}{\beta_4} \right]^{\frac{1}{2}}$$

In the next section, we are going to characterize the recovery of state feedback stability robustness in terms of c_* .

ii) The state feedback control, $f(x)$, is assumed to be Lipschitzian. We distinguish between two different cases. When the Lipschitz condition holds globally in x, we obtain a global stability result. When the Lipschitz condition holds only on a compact set, our stability result holds within a region of attraction.

Examples

1) Nonlinear Control [1]

$u = -Kx - g(x)$

such that $A - BK$ is Hurwitz and P is the unique s.p.d. solution of

$$P(A - BK) + (A - BK)' P = -I$$

$$g(x) = \begin{cases} \dfrac{B'Px}{\| B'Px \|}\Delta(\| x \|) & ; \text{if } \| B'Px \| > \eta \\ \dfrac{B'Px}{\eta}\Delta(\| x \|) & ; \textit{otherwise} \end{cases}$$

$$\Delta(\| x \|) = \frac{\delta_x + \delta_u \| K \|}{1 - \delta_u} \| x \| + K_*$$

where $K_* > \dfrac{\delta_c}{1 - \delta_u}$. Let $W = x'Px$,

case 1) If $\delta_x = \delta_u = 0$ and $\delta_c \neq 0$, then

$$\dot{W} \leq - \| x \|^2 + \frac{\delta_c^2 \eta}{2K_*}$$

Moreover the control satisfies a global Lipschitz condition.

case 2) $\delta_u = 0$, $\delta_x \neq 0$ and $\delta_c \neq 0$, then

$$\dot{W} \leq -\frac{1}{2} \| x \|^2 + 2\,\delta_c \| PB \| \| x \|$$

for sufficiently small η. Moreover the control satisfies the Lipschitz condition on a compact set in x.

2) Linear Control [2]

$$u = -Kx - \frac{1}{\eta}B'Px$$

$$W = x'Px$$

$$\dot{W} \leq -\frac{1}{2} \| x \|^2 + 2 \| B'P \| \delta_c \| x \|$$

for sufficiently small η. A linear control obviously satisfies a global Lipschitz condition.

Applying the following observer-based control to System (2.1)

$$\begin{cases} \dot{\hat{x}} = A\hat{x} - Bf(\hat{x}) + L(y - C\hat{x}) \\ u = -f(\hat{x}) \end{cases} \tag{2.3}$$

the error equation is going to be

$$\dot{e} = (A - LC)e + E\,\delta(x, u, t) \tag{2.4}$$

In this paper, we address the following problem: How should the observer gain, L, be chosen to recover the robustness properties of the state feedback design.

In order to prove the stability of the closed-loop system, we are going to form a weighted sum of $W(x)$ and a Lyapunov function candidate of the error equation. Therefore L should be chosen to ensure the existence of such a Lyapunov function with certain disturbance rejection properties. Throughout the next section, we are going to *assume* that such an observer gain exists (Assumption 3). However, in section 4 we will present sufficient conditions for its existence, as well as an algorithm for its computation.

Assumption 3: There exist similarity transformation $\Gamma(.)$, observer gain $L(.)$, Lyapunov function candidates $V_e(.)$ and $V_f(.)$, all functions of a small positive parameter ϵ, such that the following inequalities hold for every $\epsilon \in (0, \epsilon^*)$.

$$\Gamma(\epsilon)\, e = \begin{bmatrix} e_s \\ e_f \end{bmatrix}$$

$$\underline{\eta}_s \ \| e_s \|^2 \leq V_s = e_s' P_s e_s \leq \overline{\eta}_s \ \| e_s \|^2$$

$$\underline{\eta}_f \ \| e_f \|^2 \leq V_f = e_f' P_f e_f \leq \overline{\eta}_f \ \| e_f \|^2$$

$$\dot{V}_s \leq -\overline{\alpha}_s V_s$$

$$\dot{V}_f \leq -\frac{1}{\epsilon} \| e_f \|^2 + \alpha_2(\epsilon) \| e_f \| \, \| \delta(.) \| + \alpha_3(\epsilon) \| e_s \| \, \| e_f \|$$

$$0 \leq \alpha_i(\epsilon) \leq \overline{\alpha}_i \qquad i=2,3$$

$$\| \Gamma^{-1}(\epsilon) \| \leq \overline{\gamma}_h$$

where ϵ^*, α_i, $\overline{\gamma}_h$, $\overline{\alpha}_s$, $\underline{\eta}_s$, $\overline{\eta}_s$, $\underline{\eta}_f$, $\overline{\eta}_f$ are positive constants, independent of ϵ.

Remarks

i) Assumption 3 basically says that the observer has a two-time scale structure. The zero dynamics, i.e. the slow part, is unaffected by the disturbances and the fast dynamics has a disturbance rejection property.

3.Stability Results

Thm 3.1: If Assumptions 1-3 hold globally in u and x, then for every $\alpha_w > 0$, there exist non-negative constants c_x, c_e and $\overline{c}_s$, dependent on ϵ, such that for sufficiently small ϵ the closed-loop system (2.1), (2.4) is globally uniformly ultimately bounded with respect to the set Φ.

$$\Phi \triangleq \left\{ (x', e')' \in \mathbf{R}^{2n} \;\middle|\; e_s' P_s e_s \leq c_s^2 \,,\ \alpha_w W(x) + e_f' P_f e_f \leq \alpha_w \overline{\eta}_w c_x^2 + \overline{\eta}_f c_e^2 \right\}$$

where $c_s < \overline{c}_s$ can be made arbitrarily small. Moreover

$$\lim_{\epsilon \to 0} c_x(\epsilon) = c_* \tag{3.1}$$

$$\lim_{\epsilon \to 0} c_e(\epsilon) = 0 \tag{3.2}$$

Proof: Refer to Appendix A.

Thm 3.2: If Assumptions 1-3 hold globally in u and on the compact set Ψ,

$$\Psi \triangleq \left\{ x \in \mathbf{R}^n \;\middle|\; W(x) \leq c \right\}$$

where $c > \overline{\eta}_w c_*^2$, then for every $\alpha_w > 0$ there exist nonnegative constants c_x, c_e, $\overline{c}_s$ and σ_s, dependent on ϵ, such that for sufficiently small ϵ, the closed-loop system (2.1), (2.4) is uniformly ultimately bounded with respect to Φ, as given in the statement of theorem 3.1, whenever the initial condition $(x_0', e_o')' \in \Sigma$.

$$\Sigma \triangleq \left\{ (x', e')' \in \mathrm{R}^{2n} \mid e_s' P_s e_s \leq \sigma_s \,,\, \alpha_w W(x) + e_f' P_f e_f \leq \alpha_w c \right\}$$

Proof: It is similar to the proof of Theorem 3.1.

Corollary 3.3: Under the assumptions of Theorem 3.2 (Theorem 3.1), if $\delta_c = \beta_2 = \delta_u k_5 = 0$, then for sufficiently small ϵ the closed-loop system (2.1), (2.4) has a (globally) uniformly asymptotically stable equilibrium at the origin.

Remarks

i) The limiting relationships (3.1) and (3.2) imply that the projection of Φ on the x-subspace asymptotically approaches the set Ω (2.2), as ϵ tends to zero.

ii) In the corollary, requiring δ_c and β_2 to be zero implies that the state feedback controller achieves asymptotic stability, which is a reasonable assumption if we expect to achieve asymptotic stability with the observer-based control. So the corollary ensures the recovery of asymptotic stability, when $\delta_u k_5 = 0$. As a special case, one may consider the linear state feedback control, where $k_5 = 0$.

4.Observer Design

Thm 4.1: If the transfer matrix $C(sI - A)^{-1}E$ is minimum phase and left-invertible, then there exist matrices $L(.)$, $\Gamma(.)$, $P_s(.)$, $P_f(.)$ satisfying the conditions of Assumption 3.

Proof: Refer to Appendix B.

Remarks

i) The proof is constructive, providing us with an algorithm to obtain $L(.)$.

ii) Left-invertibility and minimum phase assumption is sufficient for detectibility of (C, A) [7].

Appendix A

The arguments ϵ and t are dropped, whenever it causes no confusion.

$$\dot{x} = Ax - Bf(\hat{x}) + E\delta(x, -f(\hat{x}))$$

$$\dot{e} = (A - LC)e + E\delta(x, -f(\hat{x}))$$

$$\begin{pmatrix} \dot{e}_s \\ \dot{e}_f \end{pmatrix} = \Gamma(\epsilon)(A - LC)\Gamma^{-1}(\epsilon) \begin{pmatrix} e_s \\ e_f \end{pmatrix} + \Gamma(\epsilon) E \delta(x, -f(\hat{x}))$$

let $V \triangleq \alpha_w W(x) + e_f' P_f e_f$ and $V_s \triangleq e_s' P_s e_s$.

$$\begin{aligned}\dot{V} = {} & \alpha_w \frac{\partial W}{\partial x}\Big[Ax - Bf(x) + E\,\delta(x, -f(x)) \Big] + \alpha_w \frac{\partial W}{\partial x} B \Big[f(x) - f(\hat{x}) \Big] \\ & + \alpha_w \frac{\partial W}{\partial x} E \Big[\delta(x, -f(\hat{x})) - \delta(x, -f(x)) \Big] - \frac{1}{\epsilon} \| e_f \|^2 \\ & + \alpha_2 \| e_f \| \, \| \delta(x, -f(\hat{x})) \| + \alpha_3 \| e_s \| \, \| e_f \| \qquad \text{; due to Assumption 3} \\ \le {} & -\alpha_w \beta_4 \| x \|^2 + \alpha_w(\delta_c\beta_3 + \beta_2) \| x \| + \alpha_w \delta_c \beta_1 + \alpha_w \left\| \frac{\partial W}{\partial x} B \right\| k_3 \| x - \hat{x} \| \\ & + \alpha_w \left\| \frac{\partial W}{\partial x} E \right\| k_1 k_3 \| x - \hat{x} \| - \frac{1}{\epsilon} \| e_f \|^2 \\ & + \alpha_2 \| e_f \| \Big(\delta_x \| x \| + \delta_u k_4 \| \hat{x} \| + \delta_u k_5 + \delta_c \Big) \\ & + \alpha_3 \| e_s \| \, \| e_f \| \qquad \text{; due to Assumption 1,2} \\ \le {} & -\alpha_w \beta_4 \| x \|^2 + \alpha_w(\delta_c \beta_3 + \beta_2) \| x \| + \alpha_w \delta_c \beta_1 \\ & + \alpha_w \left(\left\| \frac{\partial W}{\partial x} E \right\| k_1 k_3 + \left\| \frac{\partial W}{\partial x} B \right\| k_3 \right) \bar{\gamma}_h \left(\left(\frac{V_s}{\eta_s} \right)^{1/2} + \| e_f \| \right) \\ & - \frac{1}{\epsilon} \| e_f \|^2 + \bar{\alpha}_2 \| e_f \| \Big((\delta_x + \delta_u k_4) \| x \| + \delta_u k_5 + \delta_c \Big) \\ & + \bar{\alpha}_2 \| e_f \| \delta_u k_4 \bar{\gamma}_h \left(\left(\frac{V_s}{\eta_s} \right)^{1/2} + \| e_f \| \right) + \bar{\alpha}_3 \left(\frac{V_s}{\eta_s} \right)^{1/2} \| e_f \| .\end{aligned}$$

The following inequalities hold for some nonnegative γ_i's.

$$\begin{cases} \left\| \dfrac{\partial W}{\partial x} E \right\| k_1 k_3 + \left\| \dfrac{\partial W}{\partial x} B \right\| k_3 \le 2\gamma_1 \| x \| \\ \bar{\alpha}_2(\delta_u k_5 + \delta_c) + (\bar{\alpha}_2 \delta_u k_4 \bar{\gamma}_h + \bar{\alpha}_3) \left(\dfrac{V_s}{\eta_s} \right)^{1/2} \le \gamma_2 + \gamma_3 \sqrt{V_s} \end{cases}$$

$$\begin{aligned}\dot{V} \le {} & -\alpha_w \beta_4 \| x \|^2 + \alpha_w \left[(\delta_c \beta_3 + \beta_2) + 2\gamma_1 \bar{\gamma}_h \left(\frac{V_s}{\eta_s} \right)^{1/2} \right] \| x \| + \alpha_w \delta_c \beta_1 \\ & + 2\alpha_w \gamma_1 \bar{\gamma}_h \| x \| \, \| e_f \| - \left(\frac{1}{\epsilon} - \bar{\alpha}_2 \delta_u k_4 \bar{\gamma}_h \right) \| e_f \|^2 \\ & + \bar{\alpha}_2(\delta_x + \delta_u k_4) \| x \| \, \| e_f \| + (\gamma_2 + \gamma_3 \sqrt{V_s}) \| e_f \| .\end{aligned}$$

Let $d(\alpha_w) = \bar{\alpha}_2 \delta_u k_4 \bar{\gamma}_h + \dfrac{[\alpha_w \gamma_1 \bar{\gamma}_h + 0.5\bar{\alpha}_2(\delta_x + \delta_u k_4)]^2}{\beta_4 \alpha_w}$. It is possible to show that for

$\epsilon < \min \left\{ \dfrac{1}{d(\alpha_w)}, \epsilon^* \right\}$

there exist continuous scalar functions $C_e(\epsilon, \sqrt{V_s})$ and $C_x(\epsilon, \sqrt{V_s})$ such that

$$\dot{V} < 0 \quad when \quad \| x \| > C_x(\epsilon, \sqrt{V_s}) \quad or \quad \| e_f \| > C_e(\epsilon, \sqrt{V_s})$$

Therefore

$$\dot{V} < 0 \quad when \quad V > \alpha_w \bar{\eta}_w C_x^2(\epsilon, \sqrt{V_s}) + \bar{\eta}_f C_e^2(\epsilon, \sqrt{V_s}) \triangleq g(\epsilon, V_s) \tag{A.1}$$

Using (A.1) and the inequality $\dot{V}_s \leq -\bar{\alpha}_s V_s$ (Assumption 3), it can be seen that the system is uniformly ultimately bounded with respect to the set

$$\Phi = \left\{ (x', e')' \in \mathbf{R}^{2n} \mid V_s \leq c_1 , V \leq c_2, \ where \ c_1 > 0 \ and \ c_2 > g(\epsilon, 0) \right\}$$

The limiting relationships (3.1) and (3.2) follow easily from the expressions of $C_e(.,.)$ and $C_x(.,.)$, which are not given here due to space limitations.

Appendix B

Since $C(sI - A)^{-1}E$ is minimum phase and left-invertible, there exists nonsingular transformations Γ_1, Γ_2, Γ_3, integer $K \leq n$ and integer indices q_i, r_i, $i=1,\ldots,K$ such that

$$x = \Gamma_1 \tilde{x} \qquad y = \Gamma_2 \begin{bmatrix} \tilde{y}_f \\ \tilde{y}_s \end{bmatrix} \qquad u = \Gamma_3 \tilde{u} \qquad \delta(.) = \Gamma_3 \tilde{\delta}(.) \tag{B.1}$$

The transformation (B.1) transforms System (2.1) into the following form [for more information refer to 7]:

(From this point on, all the tilde's are dropped for notational simplicity.)

$$\begin{cases} \dot{x}_a = A_{aa} x_a + A_{af} y_f + A_{as} y_s + B_a u \\ \dot{x}_b = A_{bb} x_b + A_{bf} y_f + B_b u \\ \dot{x}_f = A_f x_f + M_f y_f + B_f u + E_f \left[D_a x_a + D_b x_b + D_f x_f + \delta(.) \right] \end{cases}$$

$$\begin{cases} y_f = C_f x_f \\ y_s = C_s x_b \end{cases}$$

where

$$\begin{aligned} A_f &= diag\ (A_{1f}, \ldots, A_{Kf}) \\ E_f &= diag\ (E_{1f}, \ldots, E_{Kf}) \\ C_f &= diag\ (C_{1f}, \ldots, C_{Kf}) \\ C_c &= diag\ (C_{1s}, \ldots, C_{Ks}) \end{aligned}$$

such that A_{if}, E_{if}, C_{if} and C_{is} have the following canonical structure.

$$A_{\mathrm{if}} = \begin{pmatrix} 0 & I_{(i-1)q_i} \\ 0 & 0 \end{pmatrix} \qquad E_{\mathrm{if}} = \begin{pmatrix} 0 \\ I_{q_i} \end{pmatrix}$$

$$C_{\mathrm{if}} = \begin{pmatrix} I_{q_i} & 0 \end{pmatrix}$$

$$C_{is} = \begin{pmatrix} I_{r_i} & 0 \end{pmatrix}$$

Moreover [Refer to Theorem 2.1, 7]

i) (A_{bb}, C_s) and $(A_{\mathrm{if}}, C_{\mathrm{if}})$ form observable pairs.

ii) Invariant zeros of (C, A, E) are the eigenvalues of A_{aa}, and hence A_{aa} is Hurwitz.

Choose the following estimator, where $\hat{x}$ is the estimate of the state x.

$$\begin{cases} \dot{\hat{x}}_a = A_{aa}\hat{x}_a + A_{af}\,y_f + A_{as}\,y_s + B_a u \\ \dot{\hat{x}}_b = A_{bb}\hat{x}_b + A_{bf}\,y_f + L_b(\,y_s - C_s\hat{x}_b\,) + B_b u \\ \dot{\hat{x}}_f = A_f\hat{x}_f + M_f\,y_f + B_f u + E_f\left[D_a\hat{x}_a + D_b\hat{x}_b + D_f\hat{x}_f\right] + L_d(\,y_f - C_f\hat{x}_f\,) \end{cases}$$

where L_b and L_d are to be chosen. Hence the error equation is going to be:

$$\begin{cases} \dot{e}_a = A_{aa}e_a \\ \dot{e}_b = (\,A_{bb} - L_b C_s\,)\,e_b \\ \dot{e}_f = (\,A_f - L_d C_f\,)e_f + E_f\,[\,D_a e_a + D_b e_b + D_f e_f\,] + E_f\,\delta(.) \end{cases}$$

Choose L_b such that $A_{bb} - L_b C_s$ is Hurwitz. In order to reject the disturbance term in the error equation, the eigenvalues of $A_f - L_d C_f$ are chosen to be fast.

Choose $L_d \triangleq diag\,(\,L_{1d}, \ldots, L_{Kd}\,)$ where

$$L_{id} \triangleq \begin{pmatrix} \dfrac{L_{i1d}}{\epsilon} \\ . \\ . \\ . \\ \dfrac{L_{iid}}{\epsilon^i} \end{pmatrix} \qquad ;L_{ijd}'s \ \textit{are}\ q_i \times q_i\ \textit{square matrices.}$$

$$\textit{such that} \quad A_i \triangleq \left(\begin{array}{c|c} -L_{i1d} & \\ . & I_{(i-1)q_i} \\ . & \\ \hline . & \\ -L_{iid} & 0 \end{array}\right) \quad \textit{is Hurwitz.}$$

Such a choice of A_i is always possible, due to the canonical structure of A_{if} and C_{if}. A scaling of e_f transforms the system into the two-time scale singularly perturbed form. Considering

$$e_{fs} \triangleq T_f\, e_f$$

$$T_f \triangleq diag\ (\ T_{1f}\,, \ldots, T_{Kf}\ )$$

$$T_{\text{if}} \triangleq diag\ (\ \frac{I_{q_i}}{\epsilon^{i-1}}\,, \frac{I_{q_i}}{\epsilon^{i-2}}\,, \ldots, I_{q_i}\ )$$

it is easy to show that

$$T_f\, E_f\ = E_f$$

$$T_f\ (\,A_f - L_d\, C_f\ )\ T_f^{-1} = \frac{1}{\epsilon}\, diag\,(A_1, \ldots, A_K) = \frac{1}{\epsilon} A_h$$

Moreover, define

$$e_s \triangleq (e_a{}', e_b{}')'$$

$$A_s \triangleq diag\,(A_{aa}\ , A_{bb} - L_b\, C_s\,)$$

$$n_s \triangleq n - \sum_{i=1}^{K} i q_i \qquad n_f\ \triangleq \sum_{i=1}^{K} i q_i$$

to obtain

$$\begin{cases} \dot{e}_s\ = A_s\, e_s \\ \dot{e}_{fs} = E_f\ (D_a, D_b\,)\, e_s + \frac{1}{\epsilon}\,(A_h + \epsilon E_f\, D_f\ T_f^{-1})\, e_{fs} + E_f\ \delta(.) \end{cases}$$

There exists an ϵ^* such that $A_h + \epsilon E_f\, D_f\, T_f^{-1}$ is Hurwitz for every ϵ on $(0, \epsilon^*)$. Choose $\epsilon \in (0, \epsilon^*)$. Let P_s and P_f be the unique symmetric positive definite solutions of the following Lyapunov equations.

$$P_s A_s + A'_s P_s = -\,I_{n_s}$$

$$P_f\ (\,A_h + \epsilon E_f\, D_f\ T_f^{-1}\,) + (\,A_h + \epsilon E_f\, D_f\ T_f^{-1}\,)'\ P_f\ = -\,I_{n_f}$$

Therefore

$$\frac{d}{dt}(\ e_s{}'\, P_s\, e_s\ ) = -\ \|\ e_s\ \|^2$$

$$\frac{d}{dt}(\ e_{fs}{}'\, P_f\ e_{fs}\ ) = 2 e_{fs}{}'\, P_f\ [\,\frac{1}{\epsilon} A_h\, e_{fs} + E_f\, D_f\ T_f^{-1} e_{fs} + E_f\ (\,D_a, D_b\,)\ e_s + E_f\ \delta(.)\,]$$

$$\leq -\frac{1}{\epsilon}\ \|\ e_{fs}\ \|^2 + 2\ \|\ P_f\, E_f\, (D_a, D_b)\ \|\ \|\ e_s\ \|\ \|\ e_{fs}\ \| + 2\ \|\ P_f\, E_f\ \|\ \|\ e_{fs}\ \|\ \|\ \delta(.)\ \| .$$

It can be easily seen that the following choice of $L(.)$, $\Gamma(.)$, V_s and V_f satisfies the requirements of Assumption 3.

$$V_s = e_s{}'\, P_s\, e_s$$

$$V_f\ = e_{fs}{}'\, P_f\ e_{fs}$$

$$\Gamma(\epsilon) = \begin{bmatrix} I & 0 \\ 0 & T_f \end{bmatrix} \Gamma_1^{-1}$$

$$L = \Gamma_1 \begin{bmatrix} A_{af} & A_{ae} \\ A_{bf} & L_b \\ M_f + L_d & 0 \end{bmatrix} \Gamma_2^{-1}$$

References

[1] M. Corless and G. Leitmann, "Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems," IEEE trans. Automat. Control, AC-26, pp. 1139-1144, 1981.

[2] J.S. Thorp and B.R. Barmish, "On guaranteed stability of uncertain linear systems via linear control," J. Optimiz. Theory Appl., vol. 35, pp. 559-579, 1981.

[3] A. Saberi and H. Khalil, "Decentralized stabilization of a class of nonlinear interconnected systems," Int. J. Control, vol. 36, pp. 803- 818, 1982.

[4] A.R. Galimidi and B.R. Barmish, "The constrained Lyapunov problem and its application to robust output feedback stabilization," IEEE trans. Automat. Control, vol. AC-31, pp. 410-419, 1986.

[5] A.R. Galimidi, "A new class of uncertain linear systems stabilizable via output feedback," Proc. CDC, Athens, 1986.

[6] C.V. Hollot and A.R. Galimidi, "Stabilizing uncertain systems: Recovering full state feedback performance via an observer," IEEE trans. Automat. Control, vol. AC-31, pp. 1050- 1053, 1986.

[7] P. Sannuti and A. Saberi, "A special coordinate basis for multivariable linear systems: Finite and infinite zero structure, squaring down and decoupling," Int. J. Control, vol. 45, pp. 1655- 1704, 1987.

CONTROL OF A PM STEPPER MOTOR USING NONLINEAR OBSERVERS AND CONTROLLERS

M. Zribi, J. Chiasson
School of Electrical Engineering
Purdue University
West Lafayette, Indiana

1. INTRODUCTION

The explosive growth of the computer industry has meant a corresponding growth for stepper motors since they provide precise positioning control for computer peripherals such as paper feed mechanisms, x-y plotters positioning and floppy disk drives [1]. Stepper motors are also widely used in commercial and military applications to do such functions as mixing, stirring and blending [1].

Our main concern here is with control aspects of the permanent magnet (PM) stepper motor. In particular we present the feedback linearization approach as in [8], [21] and the variable structure approach [10], and a multi-output observer design [14].

Stepper motors were designed to provide precise positioning control to within an integral number of steps without using sensors. They are in fact open loop stable to any step position and thus no feedback is needed to control them. However, their step responses have quite a bit of overshoot and a relatively long settling time. This motivates one to go ahead and consider feedback. Ilić et. al. [8], [21] and Taylor et. al. [9] have used the feedback linearization technique [5], [6], [7] to design a controller for a variable reluctance stepper motor. A drawback of this technique is that it requires full state feedback. In particular, position and speed sensors are required. We partially solve this problem by using a *nonlinear* observer based on the work of Krener-Respondek [14] and Krener-Isidori [23]. We show that an observer that estimates rotor speed from measurements of the phase currents and rotor position can be designed. We also show that the Krener-Respondek method [14] cannot be used to estimate position (this is true even if both phase currents and the rotor speed are measured). We propose though, that the possibility of using just the two phase currents for feedback exists. Specifically we propose to just integrate the estimated rotor speed $\hat{\omega}$ with discrete updates in position and speed at each step. Since we drive the state to equilibrium at each step along any multistep path, $\hat{\theta}$ and $\hat{\omega}$ are known at the beginning of any individual step. This in turn initializes the observer correctly for $\hat{\theta}$ and $\hat{\omega}$.

1.1. Method of Operation of the PM Stepper Motor [1]

Step motors are electromagnetic incremental-motion devices which convert digital pulse inputs to analog output motion.

The basic step motor or stepper motor consists of slotted stator equipped with two or more individual coils and a rotor with no windings. Figure 1 below shows a cutaway view of a PM stepper motor which has two sets of teeth on the rotor. Observe that the two sets of teeth are out of alignment with each other by a tooth width.

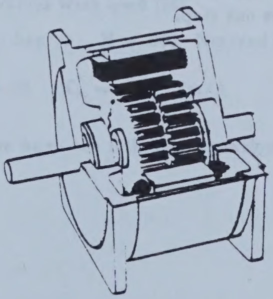
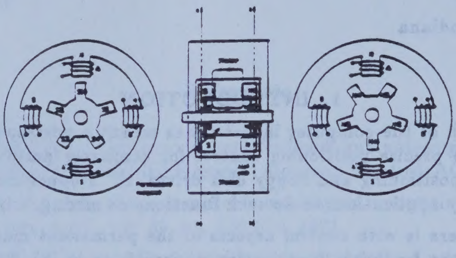


Fig. 1.

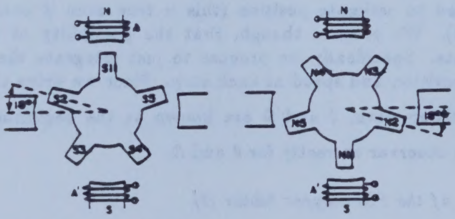
The basic construction details of the PM stepper motor are illustrated below in Fig. 2 (Del Toro [1]). The motor in Fig. 2 is shown with a four-pole stator and a five-pole rotor structure for convenience of exposition. Fig. 2.a is an axial view showing the south pole of the permanent magnet attached to the rotor shaft. Fig. 2.b is a cross sectional view showing the rotor poles of S-pole polarity; Fig. 2.c is a cross sectional view showing the north pole of the permanent magnet.



(a) (b)

Fig. 2. [1] Diagram illustrating the construction features of the PM stepper motor.

The manner in which the PM stepper motor works is described by the following diagrams below. Fig. 3 illustrates the starting point with coil (phase) A-A′ energized for a N-S orientation and coil (phase) B-B′ is de-energized. Fig. 3.a is the south end of the PM motor while Fig. 3.b is the north end.



(a) (b)

Fig. 3. [1] Starting point.

Fig. 4 below illustrates the fact that when coil A-A′ is de-energized, and coil B-B′ is energised for a N-S orientation, the rotor rotates one step.

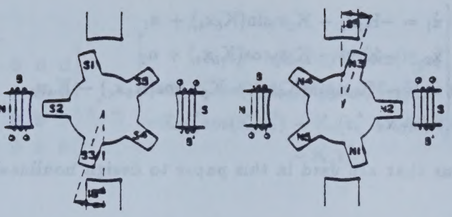


Fig. 4. [1] One step rotation of the stepper motor.

1.2 Mathematical Model of the PM Stepper Motor

The equations describing the stepper motor are [2] [3] [4] [17]:

$$\begin{cases} \dfrac{di_a}{dt} = \dfrac{1}{L}\left[v_a - Ri_a + K_m\,\omega\,\sin(N_r\theta)\right] \\ \dfrac{di_b}{dt} = \dfrac{1}{L}\left[v_b - Ri_b - K_m\,\omega\,\cos(N_r\theta)\right] \\ \dfrac{d\omega}{dt} = \dfrac{1}{J}\left[-K_m i_a \sin(N_r\theta) + K_m i_b \cos(N_r\theta) - B\omega\right] \\ \dfrac{d\theta}{dt} = \omega \end{cases} \tag{1.1}$$

i_a, i_b and v_a, v_b are the currents and voltages in phases (winding/coil) A and B respectively. Also, L and R are the self-inductance and resistance of each phase winding, K_m is the motor torque constant, N_r is the number of rotor teeth, J is the rotor inertia, B is the viscous friction constant, ω is the rotor speed and θ is the motor position.

Now, assign the state variables x_1, x_2, x_3 and x_4 such that

$$\begin{cases} x_1 = i_a \\ x_2 = i_b \\ x_3 = \omega \\ x_4 = \theta \end{cases} \tag{1.2}$$

For convenience let

$$K_1 = \frac{R}{L}\ ,\ K_2 = \frac{K_m}{L}\ ,\ K_3 = \frac{K_m}{J}\ ,\ K_4 = \frac{B}{J}\ ,\ K_5 = N_r$$

$$u_1 = \frac{1}{L}\,v_a \quad , \quad u_2 = \frac{1}{L}\,v_b$$

For design purposes the following values were used [26].

$R = 10$ ohm $L = .0011$ henrys $B = .001$ Nm/rad/s

$J = 5.7\times10^{-6}\,\mathrm{Kgm}^2$ $N_r = 50$ $K_m = .113$ Nm/A

With $\dot{x}_i = \dfrac{dx_i}{dt}$ $(i = 1,2,3,4)$ we have the following equations for the permanent-magnet stepper motor:

$$\begin{cases} \dot{x}_1 = -K_1x_1 + K_2x_3\sin(K_5x_4) + u_1 \\ \dot{x}_2 = -K_1x_2 - K_2x_3\cos(K_5x_4) + u_2 \\ \dot{x}_3 = -K_3x_1\sin(K_5x_4) + K_3x_2\cos(K_5x_4) - K_4x_3 \\ \dot{x}_4 = x_3 \end{cases} \tag{1.3}$$

These are the equations that are used in this paper to design nonlinear controllers for the PM stepper motor.

2. A NONLINEAR OBSERVER TO ESTIMATE ROTOR SPEED

2.1. Nonlinear State and Output Transformations

In this final section we show how an observer can be used to eliminate measuring the rotor speed ω (i.e., z_3). To this end let us, for convenience, consider the PM stepper motor equations given as:

$$\begin{cases} \dot{z}_1 = -K_1z_1 + K_2z_3\cos(K_5z_4) + u_1 \\ \dot{z}_2 = -K_1(z_2 + i_0) + K_2z_3\sin(K_5z_4) + u_2 \\ \dot{z}_3 = -K_3z_1\cos(K_5z_4) - K_3(z_2 + i_0)\sin(K_5z_4) - K_4z_3 \\ \dot{z}_4 = z_3 \end{cases}$$

where we have let $z_1 = x_1$, $z_2 = x_2 - i_0$, $z_3 = x_3$, and $z_4 = x_4 - \theta_s$ in 1.3. The available outputs are assumed to be the two currents z_1, z_2 and position z_4. It would then be natural to take as output $y = Cz$ where $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. It turns out though, in this particular application it is necessary to do a *nonlinear* output transformation on z_1, z_2, z_4. This is a direct consequence of theorems 2.1 and 2.2 of Krener and Respondek [14]. We take as our output

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \triangleq \begin{bmatrix} h_1(z) \\ h_2(z) \\ h_3(z) \end{bmatrix} \triangleq \begin{bmatrix} z_1 - (K_2/K_5)\sin(K_5z_4) \\ z_2 + (K_2/K_5)\cos(K_5z_4) \\ z_4 \end{bmatrix} \tag{2.1}$$

Clearly y_1, y_2, y_3 are available since z_1, z_2 and z_4 are assumed to be measurable. Now we use the following *nonlinear* state space transformation (change of coordinates)

$$\begin{aligned} z_1^* &= z_1 - K_2/K_5\sin(K_5z_4) \\ z_2^* &= z_2 + K_2/K_5\cos(K_5z_4) \\ z_3^* &= z_3 + K_4z_4 \\ z_4^* &= z_4 \end{aligned} \tag{2.2}$$

In the new coordinates the system is described by

$$\begin{aligned} \dot{z}_1^* &= -K_1z_1^* - (K_1K_2/K_5)\sin(K_5z_4^*) + u_1 \\ \dot{z}_2^* &= -K_1z_2^* + (K_1K_2/K_5)\cos(K_5z_4^*) - K_1i_0 + u_2 \\ \dot{z}_3^* &= -K_3z_1^*\cos(K_5z_4^*) - K_3z_2^*\sin(K_5z_4^*) - K_3i_0\sin(K_5z_4^*) \\ \dot{z}_4^* &= z_3^* - K_4z_4^* \end{aligned} \tag{2.3}$$

Note that the *unobservable* state z_3^* appears *linearly* in (2.3). The output equation is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1^* \\ z_2^* \\ z_3^* \\ z_4^* \end{bmatrix} \triangleq Cz^* \tag{2.4}$$

Comparing (2.1) and (2.2) it should be clear that we must have $y_1 = z_1^*$, $y_2 = z_2^*$ and $y_3 = z_4^*$. We now rewrite (2.3) as

$$\dot{z}^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} z^* + \begin{bmatrix} -K_1 z_1^* - (K_1K_2/K_5)\sin(K_5 z_4^*) \\ -K_1 z_2^* + (K_1K_2/K_5)\cos(K_5 z_4^*) - K_1 i_0 \\ -K_3 z_1^* \cos(K_5 z_4^*) - K_3(z_2^* + i_0)\sin(K_5 z_4^*) \\ -K_4 z_4^* \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ 0 \\ 0 \end{bmatrix} \tag{2.5}$$

2.2 Observer Design

Define the estimated state $\hat{z}$ to be the solution of

$$\dot{\hat{z}}^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \hat{z}^* + \begin{bmatrix} -K_1 z_1^* - (K_2K_1/K_5)\sin(K_5 z_4^*) \\ -K_1 z_2^* + (K_1K_2/K_5)\cos(K_5 z_4^*) - K_1 i_0 \\ -K_3 z_1^* \cos(K_5 z_4^*) - K_3(z_2^* + i_0)\sin(K_5 z_4^*) \\ -K_4 z_4^* \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ 0 \\ 0 \end{bmatrix} + K(y^* - \hat{y}^*) \tag{2.6}$$

where

$$K \in \mathbb{R}^{4\times 3} \qquad y^* \triangleq \begin{bmatrix} z_1^* \\ z_2^* \\ z_4^* \end{bmatrix} \qquad \hat{y}^* \triangleq \begin{bmatrix} \hat{z}_1^* \\ \hat{z}_2^* \\ \hat{z}_4^* \end{bmatrix} \tag{2.7}$$

Observation Note that the second term on the right hand side of equation (2.6) (which contains the nonlinearities) is defined in terms of the *measured* output quantities z_1^*, z_2^*, z_4^* *not* the estimated ($\hat{z}$) states.

Now the whole point of the above transformation is that (see [14]) we can use the differential equations for z^* and $\hat{z}^*$ to get a differential equation for the error $e^* \triangleq z^* - \hat{z}^*$ which is *linear*. Specifically we see that $z^* - \hat{z}^*$ satisfies

$$\dot{z}^* - \dot{\hat{z}}^* = (A - KC)\,(z^* - \hat{z}^*) \tag{2.8}$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

We choose K so that A − KC is stable. In particular we take K to be the form

$$K = \begin{bmatrix} k_{10} & 0 & 0 \\ 0 & k_{20} & 0 \\ 0 & 0 & k_{30} \\ 0 & 0 & k_{31} \end{bmatrix} \tag{2.10}$$

Since the above system is in observer canonical form it is an easy task to arbitrarily assign the spectrum of A − KC.

Below we give the nonlinear observer transformed back to the original coordinates.

$$\dot{\hat{z}}_1 = -K_1 z_1 + K_2 \hat{z}_3 \cos(K_5 \hat{z}_4) + k_{10}(z_1 - \hat{z}_1) + k_{10}(K_2/K_5)(\sin(K_5 z_4) - \sin(K_5 \hat{z}_4))$$

$$+ (K_2K_4 + K_2k_{31})(z_4 - \hat{z}_4)\cos(K_5 \hat{z}_4) + u_1$$

$$\dot{\hat{z}}_2 = -K_1z_2+K_2\hat{z}_3\sin(K_5\hat{z}_4)-K_1i_0+k_{20}(z_2-\hat{z}_2)+k_{20}(K_2/K_5)(\cos(K_5z_4)-\cos(K_5\hat{z}_4)) \tag{2.11}$$

$$+(K_2K_4+K_2k_{31})(z_4-\hat{z}_4)\sin(K_5\hat{z}_4)+u_2$$

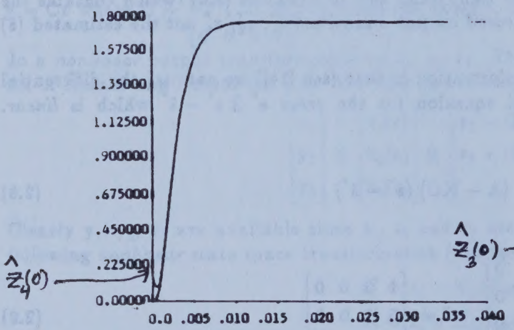
$$\dot{\hat{z}}_3 = -K_3z_1\cos(K_5z_4)-K_3z_2\sin(K_5z_4)-K_3i_0\sin(K_5z_4)+(K_4^2-K_4k_{31}+k_{30})(z_4-\hat{z}_4)-K_4\hat{z}_3$$

$$\dot{\hat{z}}_4 = \hat{z}_3 + (-K_4+k_{31})(z_4-\hat{z}_4)$$

Remark Note that the observer *does not* depend on z_3 as was purposely designed and thus may be implemented. Also note that this observer is defined globally so that $\hat{z}(0)$ may be completely arbitrary and still have convergence of $\hat{z}(t)$ to $z(t)$.

2.3 Simulation and Results

The above observer was implemented into the simulation along with the stepper motor equations and the variable structure controller. The observer's error dynamics (i.e., $A - KC$) were designed to have poles at 1450, 1500, 1550 and 1600. The initial condition of the observer (2.11) was set to $(\hat{z}_1(0), \hat{z}_2(0), \hat{z}_3(0), \hat{z}_4(0)) = (.35, .4, 1.5, .2,)$ while the system (3.12) had its initial state set to $(z_1(0), z_2(0), z_3(0), z_4(0)) = (.75, 0, 0, 0)$. These initial conditions allow us to study how well the observer tracks the actual state. The simulation results are shown below.



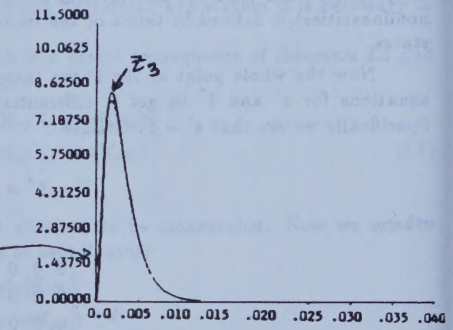


Fig. 2.1a
z_4 and $\hat{z}_4$
$z_4(0) = 0$, $\hat{z}_4(0) = .2$

Fig. 2.1b
z_3 and $\hat{z}_3$
$z_3(0) = 0$, $\hat{z}_3(0) = 1.5$

2.4 Observer Based on Measurements of i_a, i_b

It can be shown that it is not possible to use the Krener-Respondek method to design an observer based on measurements of i_a, i_b and ω. Clearly then, one could not design an observer using the above method with only measurements of i_a and i_b. We propose though, that the possibility of using just the two phase currents for feedback may work by just integrating the estimated speed $\hat{z}_3$ to get an estimate $\hat{z}_4$ of position. This is operating "open loop" but if the system is working in a single step mode (e.g., a line printer) then the initial speed is zero and the initial position is known. Since these initial conditions are known it seem plausible that this combination of open loop and feedback may work. Further, since the stepper motor system is stable, one could turn off the feedback in a neighborhood of the final position and let the open loop system dynamics take over which would then eliminate any position bias.

Remark See Appendix A for the transformation to observer form for stepper motor system described by (1.12).

CONCLUSIONS

We have shown how non-linear methods- feedback linearization, variable structure control, and Krener-Respondek observers can be used to control a practical system, i.e., a PM stepper motor.

ACKNOWLEDGEMENTS

Both authors would like to extend their sincere thanks to Dr. Stanislaw H. Żak for his helpful discussions and comments.

REFERENCES

[1] V. Deltoro, *Electric Machines and Power Systems,* Prentice-Hall, 1985

[2] B. C. Kuo, *Automatic Control Systems,* Prentice-Hall, 1982.

[3] B. C. Kuo, *Theory and Applications of Step Motors,* West Publishing Co., 1974.

[4] B. C. Kuo, J. Tal, *Incremental Motion Control, Step Motors and Control Systems,* Vol. II, SRL Publishing, Champaign, IL, 1978.

[5] R. Su, G. Meyer, and L. R. Hunt, "Design for multi-input nonlinear systems," in *Differential Geometric Control Theory,* edited by R. W. Brochett, R. S. Millman and H. J. Sussman, Birkhauser, Boston, pp. 268-298, 1983.

[6] R. Sommer, "Control design for multivariable nonlinear time varying systems," Int. J. Control. Vol. 31, No. 5, pp. 883-891, 1980.

[7] S. H. Żak and C. A. MacCarley, "State feedback control of nonlinear systems," Int. J. Control, Vol. 43, No. 5, pp. 1497-1514, 1986.

[8] M. Ilić-Spong, R. Marino, S. Peresada, and D. G. Taylor, "Nonlinear control switched reluctance motors in robotics applications," Applied Motion Control Conference, June 1986, Mpls. Mn.

[9] D. G. Taylor, M. Ilić-Spong, S. Peresada, "Nonlinear composite control of switched reluctance motors," 1986 ACC, Seattle.

[10] V. I. Utkin, *Sliding modes and their application in variable structure systems,* MIR Publishers, Moscow, 1978.

[11] R. Marino, "High gain feedback in nonlinear control systems," Int. Journal of Control, 1985, Vol. 42, No. 6, pp. 1369-1385.

[12] L. R. Hunt, M. Luksic, R. Su, "Exact linearization of input-output systems," Int. J. Control, 1986, Vol. 43, No. 1, 247-255.

[13] B. L. Walcott and S. H. Żak, "Observation of dynamical systems in the presence of bounded nonlinearities/uncertainties," Proc. 25th CDC, Athens, Greece, Dec. 1986.

[14] A. J. Krener and W. Respondek, "Nonlinear Observers with Linearizable Error Dynamics," SIAM J. Control and Optimization, Vol. 23, No. 2, pp. 197-216, 1985.

[15] V. I. Utkin, Sliding Modes and Their Application in Variable Structure Systems, MIR Publishers, Moscow, 1978.

[16] U. Itkis, Control Systems of Variable Structure, J. Wiley and Sons, Inc., New York, 1976.

[17] T. Kenjo, Stepping Motors and Their Microprocessor Controls, Clarendon Press, Oxford, 1984.

[18] P. P. Acarnley, Stepping Motors: A guide to modern theory and practice, P. Peregrinus Ltd., Stevenage, UK, 1982.

[19] R. S. Millman and G. D. Parker, *Elements of Differential Geometry,* Prentice Hall, 1977.

[20] T. Kailath Linear Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.

[21] M. Ilić-Spong, R. Marino, S. M. Peresada, and D. G. Taylor, "Feedback Linearising Control of Switched Reluctance Motors," IEEE A.C. May, 1987.

[22] M. Zribi, "Control of a Permanent Magnet Stepper Motor Using Modern Nonlinear Control Techniques," M.S. Thesis, Purdue University, West Lafayette, IN, May 1987.

[23] A. J. Krener and A. Isidori, "Linearization by Output Injection and Nonlinear Observers," Systems and Control Letters, 3 (1983), pp. 47-52.

[24] B. L. Walcott, M. J. Corless and S. H. Żak, "Comparative Study of Nonlinear State Observation Techniques," Int. J. Contr., to appear 1987.

[25] M. Zribi and J. Chiasson, "Control of a PM Stepper Motor Using Modern Nonlinear Control Techniques," Proceedings Conference on Applied Motion Control, June 1987.

[26] G. White, "Open-loop Step Motor Controls for Print-Wheel Drives," in Proc. 9th Annual Symposium on Incremental Motion Control Systems and Devices, B.C. Kuo, Editor, June 1980.

APPENDIX A

Derivation of the Observer Transformation

The results in [14] are quite technical and do require a knowledge of abstract manifold theory. The book by Millman and Parker [19] does an excellent job of bridging the gap between the classical differential geometry of surfaces and abstract manifold theory.

Using the notation of [14] we write

$$\xi_{11} \triangleq \psi_1 = z_4$$

$$\dot{\xi}_{11} \triangleq \xi_{12} = L_f(z_4) = z_3$$

$$\dot{\xi}_{12} = L_f(z_3) = -K_3 z_1 \cos(K_5 z_4) - K_3(z_2 + i_0)\sin(K_5 z_4) - K_4 z_3$$

$$\xi_{21} = \psi_2 = z_1$$

$$\dot{\xi}_{21} = L_f(z_1) = -K_1 z_1 + K_2 z_3 \cos(K_5 z_4) + u_1$$

$$\xi_{31} = \psi_3 = z_2$$

$$\dot{\xi}_{31} = L_f(z_2) = -K_1(z_2 + i_0) + K_2 z_3 \sin(K_5 z_4) + u_2$$

We invert the transformation

$$\xi_{11} = z_4\,,\quad \xi_{12} = z_3\,,\quad \xi_{21} = z_1\,,\quad \xi_{31} = z_2$$

and substitute to get

$$\xi_{11} = z_4 = \psi_1$$

$$\dot{\xi}_{11} = \xi_{12} \tag{A1}$$

$$\dot{\xi}_{12} = -K_3\xi_{21}\cos(K_5\xi_{11}) - K_3(\xi_{31} + i_0)\sin(K_5 z_{11}) - K_4\xi_{12} \triangleq f_1(\xi) \tag{A2}$$

$$\xi_{21} = z_1 = \psi_2$$

$$\dot{\xi}_{21} = -K_1\xi_{21} + K_2\xi_{12}\cos(K_5\xi_{11}) + u_1 \triangleq f_2(\xi) + u_1 \tag{A3}$$

$$\xi_{31} = z_2 = \psi_3$$

$$\dot{\xi}_{31} = -K_1(\xi_{31} + i_0) + K_2\xi_{12}\sin(K_5\xi_{11}) + u_2 \triangleq f_3(\xi) + u_2 \tag{A4}$$

In these coordinates the system is in observable form with observability indices $k_1 = 2$, $k_2 = 1$, $k_3 = 1$ [14]. However, $f_2(\xi)$ and $f_3(\xi)$ are *not* in *special observable form* [14]. To bring the system into *special observable form* we use Remark 2.4 and Theorem 2.2 of [14].

We find the vectors Y^j by solving

$$L_{Y^j}(\psi_i) = <d\psi_i, Y^j> = \begin{cases} 1 & \text{if} \quad i = j \\ f_{i;jk_i+1} & \text{if} \quad k_j > k_i \\ 0 & \text{otherwise} \end{cases}$$

where $f_{i;jk_i+1} \triangleq \dfrac{\partial f_i}{\partial \xi_{1k_i+1}}$. In particular for $i = 2$, $j = 1$, $k_2 + 1 = 2$ we have

$$f_{2;12} = \frac{\partial f_2}{\partial \xi_{12}} = K_2 \cos(K_5 \xi_{11}) = K_2 \cos(K_4 \psi_1)$$

and for $i = 3$, $j = 1$, $k_3 + 1 = 2$ we have

$$f_{3;12} = \frac{\partial f_3}{\partial \xi_{12}} = K_2 \sin(K_5 \xi_{11}) = K_2 \sin(K_5 \psi_1)$$

and $f_{i;jk_i+1} = 0$ for all other i,j. We then solve [14]

$$\begin{bmatrix} d\psi_1 \\ d\psi_2 \\ d\psi_3 \end{bmatrix} \begin{bmatrix} Y^1(\psi) \; Y^2(\psi) \; Y^3(\psi) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ K_2\cos(K_5\psi_1) & 1 & 0 \\ K_2\sin(K_5\psi_1) & 0 & 1 \end{bmatrix}$$

where

$$d\psi_1 = [1 \;\; 0 \;\; 0] \,, \; d\psi_2 = [0 \;\; 1 \;\; 0] \,, \; d\psi_3 = [0 \;\; 0 \;\; 1]$$

The solution is

$$Y^1(\psi) = [1 \;\; K_2\cos(K_5\psi_1) \;\; K_2\sin(K_5\psi_1)]^T$$

$$Y^2(\psi) = [0 \;\; 1 \;\; 0]^T$$

$$Y^3(\psi) = [0 \;\; 0 \;\; 1]^T$$

A change of output coordinates given by

$$\tilde{\psi}_1(\psi) = \tilde{\psi}_1(\psi_1, \psi_2, \psi_3)$$

$$\tilde{\psi}_2(\psi) = \tilde{\psi}_2(\psi_1, \psi_2, \psi_3)$$

$$\tilde{\psi}_3(\psi) = \tilde{\psi}_3(\psi_1, \psi_2, \psi_3)$$

will put the system into special observable form [14] if

$$L_{Y^1}(\tilde{\psi}_i) = 0 \qquad i = 2,3$$

Explicitly we write

$$\begin{bmatrix} \dfrac{\partial \tilde{\psi}_2}{\partial \psi_1} \,, & \dfrac{\partial \tilde{\psi}_2}{\partial \psi_2} \,, & \dfrac{\partial \tilde{\psi}_2}{\partial \psi_3} \end{bmatrix} \begin{bmatrix} 1 \\ K_2\cos(K_5\psi_1) \\ K_2\sin(K_5\psi_1) \end{bmatrix} = 0 \quad \text{and} \quad \begin{bmatrix} \dfrac{\partial \tilde{\psi}_3}{\partial \psi_1} \,, & \dfrac{\partial \tilde{\psi}_3}{\partial \psi_2} \,, & \dfrac{\partial \tilde{\psi}_3}{\partial \psi_3} \end{bmatrix} \begin{bmatrix} 1 \\ K_2\cos(K_4\psi_1) \\ K_2\sin(K_5\psi_1) \end{bmatrix} = 0$$

A solution is given by

$$\tilde{\psi}_1 = \psi_1$$

$$\tilde{\psi}_2 = \psi_2 - K_2/K_5 \sin(K_5\psi_1)$$

$$\tilde{\psi}_3 = \psi_3 + K_2/K_5 \cos(K_5\psi_1)$$

The system is then in special observable form as shown below.

$$\xi_{11} = \psi_1(z) = z_4 \quad \xi_{12} \triangleq L_f\psi_1 = z_3$$

$$\dot{\xi}_{11} = \xi_{12}$$

$$\dot{\xi}_{12} = -K_3\left[\xi_{21} + K_2/K_5\sin(K_5\xi_{11})\right]\cos(K_5\xi_{11})$$

$$-K_3\left[\xi_{31} - K_2/K_5\cos(K_5\xi_{11}) + i_0\right]\sin(K_5\xi_{11}) - K_4\xi_{12}$$

$$= -K_3\xi_{21}\cos(K_5\xi_{11}) - K_3(\xi_{31} + i_0)\sin(K_5\xi_{11}) - K_4\xi_{12} \triangleq f_1(\xi)$$

$$\xi_{21} = \psi_2(z) = z_2 + (K_2/K_5)\sin(K_5z_4)$$

$$\dot{\xi}_{21} = L_f\psi_2 = -K_1\left[\xi_{21} + (K_2/K_5)\sin(K_5\xi_{11})\right] + u_1 \triangleq f_2(\xi) + u_1$$

$$\xi_{31} = \psi_3(z) = z_2 + (K_2/K_5)\cos(K_5z_4)$$

$$\dot{\xi}_{31} = L_f\psi_3 = -K_1\left[\xi_{31} - (K_2/K_5)\cos(K_5\xi_{11}) + i_0\right] + u_2 \triangleq f_3(\xi) + u_2$$

The following state space transformation then puts the system into observer form

$$x_{11} = \xi_{11} \qquad x_{21} = \xi_{21}$$
$$x_{12} = \xi_{12} + K_4\xi_{11} \qquad x_{31} = \xi_{31}$$

The overall transformation is given by equation (2.2) where $z_1^* = x_{21}$, $z_2^* = x_{31}$, $z_3^* = x_{12}$ and $z_4^* = x_{11}$

KHARITONOV'S THEOREM: IS THERE A GENERALIZATION FOR DEPENDENT COEFFICIENT VARIATIONS?

by

B. Ross Barmish

Department of Electrical and Computer Engineering

University of Wisconsin-Madison

Madison, Wisconsin 53706

Summary

Kharitonov's Theorem [1] on the stability of interval polynomials has provided much impetus for recent research. From a control theoretic point of view, however, this theorem has one fundamental limitation: The coefficient perturbations are assumed to be *independent* whereas in applications, perturbations in the physical parameters of a linear system almost always lead to *dependent* perturbations in the coefficients of the characteristic polynomial. Therefore, "robustness margins" derived from Kharitonov's Theorem tend to be conservative because the use of an interval polynomial is tantamount to treating dependent coefficient perturbations as if they are independent.

In view of this limitation, much recent research has concentrated on the case of it linearly dependent coefficient perturbations—the first level of generalization beyond Kharitonov's interval polynomial framework. If a polynomial's coefficients depend linearly on underlying physical parameters which are known only within given bounds, then it is easy to verify that the resulting family of polynomials is a convex polytope $\mathcal{P}$ generated by finitely many *extreme polynomials* $p_1(s), p_2(s), \ldots, p_\ell(s)$. Within this framework, a rather general result is established by Bartlett, Hollot and Lin [2]. Namely, in order for all polynomials in $\mathcal{P}$ to have their zeros in a simply connected region $\mathcal{D}$, it is neccessary and sufficient that the polynomials associated with the exposed edges of $\mathcal{P}$ have all their zeros in $\mathcal{D}$. Noting that each edge of a polytope depends on only 2 extreme points, the stability problem is reduced to finitely many (perhaps large in number) of one dimensional problems; e.g., for each combination of i and j, determine if $\lambda p_i(s) + (1-\lambda)p_j(s)$ has all its zeros in $\mathcal{D}$ for all $\lambda \in [0,1]$. For the case when $\mathcal{D}$ is the left half plane, this result can be refined to eliminate the λ-sweep leading to a "one-shot" edge stability test; see Bialas [3] and Fu and Barmish [4].

One important motivation for the current work is derived from the fact that existing results on zero location in a region $\mathcal{D}$ do not generalize Kharitonov's Theorem. More specifically, for the special case when the polytope $\mathcal{P}$ reduces to Kharitonov's multi-dimensional rectangle, the edge computations associated with [2]-[4] do *not* simplify. Hence, one is left wondering whether some other polytope stability criterion can be given which generalizes Kharitonov's result and avoids the combinatorics of edges. In the full version of this paper [5], we answer this question the affirmative. To this end, the following result is established: Associated with the polytope $\mathcal{P}$ and the region $\mathcal{D}$ are four real continuous functions $K_1(\delta), K_2(\delta), K_3(\delta)$ and $K_4(\delta)$ of a real variable δ and a bounded interval Δ having the following property: All polynomials in $\mathcal{P}$ have their zeros interior to $\mathcal{D}$ if and only if, for each $\delta \in \Delta$, at least one of the $K_i(\delta)$ is positive.

REFERENCES

[1] Kharitonov, V. L., *Asymptotic Stability of an Equilibrium Position of a Family of Systems of Linear Differential Equations*, Differential'nye Uraveniya, Vol. 14, no. 11, pp. 1483 - 1485, 1978.

[2] Bartlett, A. C., Hollot, C. V., and Lin, H., *Root Locations of an Entire Polytope of Polynomials: It Suffices to Check the Edges*, Proceedings of the American Control Conference, Minneapolis, 1987. Also in press for Mathematics of Control, Signals and Systems.

[3] Bialas, S,, *A Necessary and Sufficient Condition for the stability of Convex Combinations of Stable Polynomials or Matrices*, Bulletin of the Polish Academy of Sciences, Technical Sciences, Vol. 33, no. 9-10, pp. 473-480, 1985.

[4] Fu, M. and Barmish, B. R., *Stability of Convex and Linear Combinations of Polynomials and Matrices Arising in Robustness Problems*, Proceedings of the Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, 1987.

[5] Barmish, B. R., *Four Kharitonov-like functions for Robust Stability Problems with Dependent Coefficient Perturbations*, in preparation.

Acknowledgement: This work was supported by the National Science Foundation under Grant No. ECS-8611728.

EIGENVALUE BOUNDS FOR THE SOLUTION OF DISCRETE STOCHASTIC RICCATI EQUATION

ENGIN YAZ
Electrical Engineering Department
University of Arkansas
Fayetteville, AR 72701

INTRODUCTION

The following algebraic Riccati-like matrix equation (which will be called the discrete stochastic Riccati equation)

$$P = \overline{a^2}\, A^T PA - \overline{ab}^2\, A^T PB\, (R + \overline{b^2} B^T PB)^{-1}\, B^T PA + Q \tag{1}$$

arises in finding the optimal feedback control that minimizes the infinite horizon quadratic cost functional

$$J = E\{\sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k\}, \quad Q = Q^T > 0,\ R = R^T > 0 \tag{2}$$

subject to the n-dimensional stochastic system constraint

$$x_{k+1} = a_k(\omega) A x_k + b_k(\omega) B u_k \tag{3}$$

where $a_k(\omega)$ and $b_k(\omega)$ are scalar white noise sequences with stationary statistics, which may be correlated with each other at each time instant k, and $A \varepsilon R^{n\times n}$ and $B \varepsilon R^{n\times m}$ are constant matrices.

Our aim in this work is to find both upper and lower bounds for the extreme eigenvalues of the solution matrix P to the discrete stochastic Riccati equation (1). The motivations behind this search for bounds are provided by our work related to the numerical solution of (1) and the robustness study of the optimal solution in this stochastic problem. A good survey of results for deterministic Riccati equations can be found in [1] and [2].

To obtain a unique, positive definite symmetric solution in (1), we will assume that the uncertainty threshold [3] is not exceeded, or for a square, full rank B in (3), we have

$$[\overline{a^2} - (\overline{ab}^2/\overline{b^2})]^{\frac{1}{2}}\, \rho(A) < 1 \tag{4}$$

where $\rho(.)$ denotes the spectral radius. Positive definite, symmetric solution of (1) allows the ordering of the eigenvalues of P as follows:

$$\lambda_1(P) \geq \lambda_2(P) \geq \ldots \geq \lambda_n(P) \tag{5}$$

MAIN RESULTS

In this work, we find bounds on the extreme eigenvalues $\lambda_1(P)$ and $\lambda_n(P)$ in terms of the eigenvalues of the system and weighting matrices and the noise characteristics. We start with a lower bound on the maximum eigenvalue.

Result 1. The maximum eigenvalue $\lambda_1(P)$ of the solution P to (1) satisfies

$$\lambda_1(P) \geq \{\beta_1 + [\beta_1^2 + 4\varepsilon\lambda_1(S)\lambda_n(Q)]^{\frac{1}{2}}\}/2\varepsilon\lambda_1(S) \tag{6}$$

where

$$\beta_1 = (\overline{ab}^2/\overline{b^2})\rho^2(A) - \varepsilon + \lambda_1(S)\lambda_n(Q),\quad S = \overline{b^2} BR^{-1}B^T,\quad \varepsilon = 1 - \rho^2(A)[\overline{a^2} - (\overline{ab}^2/\overline{b^2})] \tag{7}$$

Result 2. $\lambda_n(P)$ satisfies

$$\lambda_n(P) \leq \{\beta_2 + [\beta_2^2 + 4\varepsilon\lambda_n(S)\, \lambda_1(Q)]^{1/2}\}/2\varepsilon\, \lambda_n(S) \tag{8}$$

where

$$\beta_2 = (\overline{ab}^2/\overline{b^2})\gamma_n^2 + \lambda_n(S)\, \lambda_1(Q) - \varepsilon, \; \gamma_n = \min_i |\lambda_1(A)| \tag{9}$$

The next result is on an upper bound on the maximum eigenvalue of P

Result 3. $\lambda_1(P)$ satisfies

$$\lambda_1(P) \leq 2\lambda_1(Q)/ \{\beta_3 + [\beta_3^2 + 4\lambda_1(Q) - \lambda_n(S)(1-\psi\lambda_1(A^TA)]^{1/2}\} \tag{10}$$

where

$$\beta_3 = 1-\overline{a^2}\, \lambda_1(A^TA) - \lambda_1(Q)\lambda_n(S), \; \psi = [\overline{a^2} - (\overline{ab}^2/\overline{b^2})] \tag{11}$$

This bound requires that $\psi\, \lambda_1(A^TA) < 1$ which is stronger than (4).

The last result gives a lower bound on the minimum eigenvalue of P.

Result 4. $\lambda_n(P)$ satisfies

$$\lambda_n(P) \geq \{\beta_4 + [\beta_4^2 + \lambda_1(S)[1-\psi\lambda_n(A^TA)]\, \lambda_n(Q)]^{1/2}\}/2\lambda_1(S)[1-\psi\lambda_n(A^TA)] \tag{12}$$

where

$$\beta_4 = 1 - \overline{a^2}\, \lambda_n(A^TA) - \lambda_n(Q)\lambda_1(S) \tag{13}$$

In this result, $\psi\lambda_n(A^TA) < 1$ is guaranteed by (4).

The proofs of these results involve the applications of mathematical tools in [4] and omitted here due to space restrictions. Since these results are first ones of their kind, comparison with the alternatives is not possible. We would also like to remark here that the proposed bounds do not generally entail more restrictions than are usually put on the system and noise parameteres to secure a unique, positive definite solution of (1).

REFERENCES

[1] T. Mori and I.A. Derese, "A brief summary of the bounds on the solution of the algebraic matrix equations in control theory," Int. J. Control, vol. 39, pp. 247-256, 1984.

[2] E. Yaz, "Bounds for the eigenvalues of the solution matrix of the algebraic Riccati equation," Int. J. Systems Sci., vol. 16, pp. 815-820, 1985.

[3] R. T. Ku and M. Athans, "Further results on the uncertainty threshold principle," IEEE Trans. Autom. Control, vol. AC-22, pp. 866-868, 1977.

[4] M. Marcus and H. Minc , A Survey of Matrix Theory and Matrix Inequalities, Allyn and Bacon, Inc., Boston, MA, 1964.

ROBUST FLIGHT CONTROL SYSTEM DESIGN TECHNIQUE USING GAIN SPACE WITH PERFORMANCE AND HANDLING QUALITIES CONSTRAINTS

GEORGE AZIZ AND JOHN HO
The Boeing Commercial Airplane Company, Seattle, Washington

INTRODUCTION

Parameter space design as formulated by J. Ackermann (ref. 1, 2) maps design objectives in the s-plane to the space of the state feed back gains.

This paper shows that the concept of gain space boundaries can be used as a constraint to the optimization problem and also addresses other requirements such as covariance constraints on states and outputs. Sensor delays and anti-aliasing requirements are also addressed.

To select an optimal gain within the gain space, a non-linear optimization program, "Sandy", was used (ref. 3). These gain values are determined from the minimization of a quadratic performance index for a linear time invariant system subjected to random initial conditions or disturbances. It will be shown in this paper that when gain space is incorporated into Sandy, less computation is needed for a design.

Four flight conditions were used in the design. The dynamic pressure ranges from 227 psf to 462 psf which includes climb, cruise and descent conditions.

The control law is designed to meet specific requirements in the area of performance, flying qualities and robustness. This is summarized in Table 1.

GAIN SPACE FOR FLYING REQUIREMENT

The controller structure is assumed to be $U = -\underline{\kappa}^T\underline{X}$, where $\underline{\kappa} = [k_1 \ldots k_4]$ is the feedback gain vector and $\underline{X}$ is the state vector $[\theta\theta \ \ \alpha \ u]$. For the design of an Attitude Control for an Autopilot, the relationship between K_1 and K_2 is determined with K_3 and K_4 set to zero. For a two dimensional problem the solution is obtained from $(k_1, k_2) = [(k_b{}^T - t^T E_b)(SE_b)^{-1}S + t^T]\ Ea$.

The intersection of the gain space of all four flight conditions is shown in Figure 1. Selecting a point (-3, -1) with the gain space by designer choice yields the result shown in Table 2.

Sensor anti-aliasing filters or time delays can be modeled by first or second order filters and the gain space can be obtained for the system that is modified by the introduction of these filters. Figure 2 shows the modified gain space altered by a second order filter on Attitude Signal for flight condition 1. It shows the loss of territory close to the damping line when we include the sensor anti-aliasing filter as expected.

SANDY DESIGN BY INCORPORATING GAIN SPACE

The gain space shown in Figure 1 was incorporated into Sandy in terms of linear and direct constraints. The design solution was obtained using fewer iterations because the gain space restricted the area of the search by the program. This particular example took 20% less iterations and yet the solution was in the vicinity of the point (-3, -1).

The point (-3, -1), however, did not satisfy the performance requirements, i.e. the normal acceleration response exceeded 1 g. under moderate turbulence conditions. Hence, Sandy was used with the weighting of the performance index chosen so that it would minimize normal acceleration response under turbulence conditions. The gains were found to be -.7, -.2 and analysis showed that the design requirements were satisfied.

GAIN SPACE WITH COVARIANCE CONTROL

Much work has been done on covariance control theory. Subject to controllability, the covariance gain is $\underline{K}$: $-\frac{1}{2}B + [\Gamma\omega\Gamma^T + AX + XA^T - (TS_K T^T)]\ X^{-1}$. For a single input system the gain solution that satisfies a certain covariance matrix is unique (ref. 4).

To obtain a solution a covariance matrix X is required. The diagonal terms are specified as they represent the state RMS quantity in the presence of turbulence. Some cross covariance term relationships can be obtained by taking a series of expected values of $[X_i \dot{X}_i]$.

In our example, since position and rate of pitch attitude are uncorrelated at a given time, i.e. $\sigma\theta\theta = 0$, hence, $a_{11}\sigma\theta^2 + a_{13}\sigma\theta u + a_{14}\sigma\theta\alpha + b_1\sigma\theta\delta_e = 0$. For some prescribed $\sigma\theta^2$, $\sigma\theta\delta_e$, the relationship between $\sigma\theta\alpha$, $\sigma\theta u$ is obtained.

By varying elements in the covariance matrix X from σx_i^2 (min) to σx_i^2 (max) together with the cross covariance relationship, the designer can partition the gain space with lines representing different covariance response. This will give the designer additional trade information when selecting operating points.

It was demonstrated that gain space can complement and enhance other conventional design techniques. A 2-D or 3-D graphic representation to outline spaces to meet requirements such as stability robustness in light of plant variations, performance margins and robustness against sensor failures will provide a practical tool to design engineers and facilitate the design process.

REFERENCES

1. "Robust Flight Control: A Design Example", S. Norman Franklin and J. Ackermann, Guidance and Control, Dec. 1981
2. "Parameters Space Design of Robust Control Systems", J. Ackermann, IEEE Transactions on Automatic Control AC-25, Dec. 1980, pp. 1058-1072.
3. "A Design Algorithm for Robust Low Order Controllers", U. Ly, Report No. 536, November, 1982, Dept. of Aero and Astro, Stanford University
4. "Covariance Control Theory" A.F. Hotz and R.E. Skelton, International Journal of Control, Vol. 46, No. 1, 1987, pp. 13-32

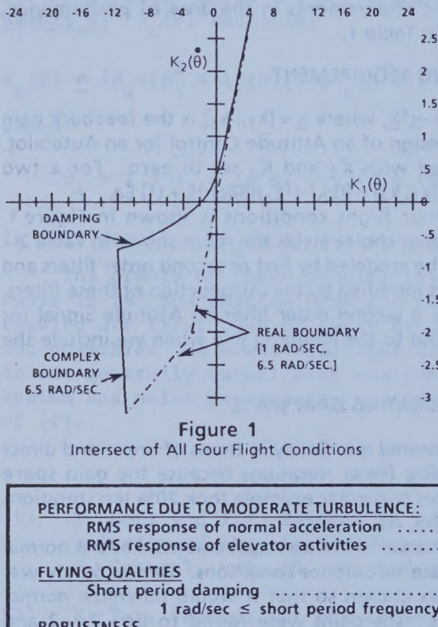


Figure 1

Intersect of All Four Flight Conditions

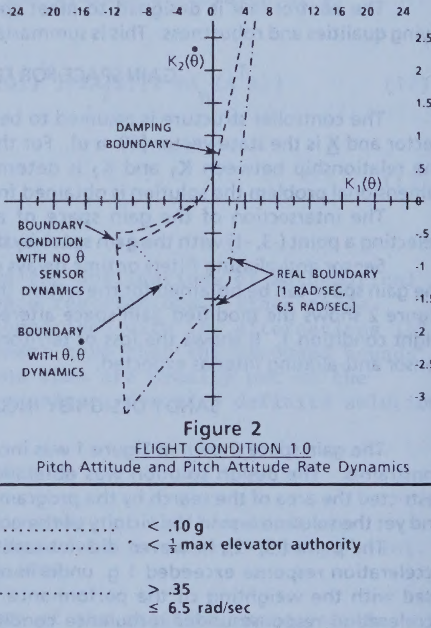


Figure 2

FLIGHT CONDITION 1.0

Pitch Attitude and Pitch Attitude Rate Dynamics

PERFORMANCE DUE TO MODERATE TURBULENCE:	
RMS response of normal acceleration	< .10 g
RMS response of elevator activities	< $\frac{1}{3}$ max elevator authority
FLYING QUALITIES	
Short period damping	> .35
1 rad/sec ≤ short period frequency	≤ 6.5 rad/sec
ROBUSTNESS	
Gain margin	≥ 6 db
Phase margin	≥ 45 deg
Control loop gain	≤ – 10 db beyond 10 rad/sec

TABLE 1

FLIGHT CONDITION	ALTITUDE	QC	GAIN	SHORT PERIOD FREQUENCY	DAMPING	FLYING QUALITY
1			(–3, –1)			Satisfactory
2			(–3, –1)	3.9089	0.67034	Satisfactory
3			(–3, –1)	4.2729	0.714	Satisfactory
4			(–3, –1)	3.6006	0.59942	Satisfactory

TABLE 2 – Summary of Flying Quality Results

SOME RESULTS ON THE ROLE OF 2-ALTERNATING UPPER PROBABILITIES IN SIGNAL-PLUS-NOISE MODELS

AMIR SADROLHEFAZI, TERRENCE L. FINE
School of Electrical Engineering, Cornell University, Ithaca, NY 14850

We consider scalar signal-plus-noise models that are robust with respect to both the signal amplitudes and the noise distribution. The signal uncertainty is modelled by requiring the signal amplitudes to fall into one of two subsets of the real line (the shift sets), while the noise contamination can be modelled by such commonly used neighborhood models as ε-contamination, total variation, etc. In statistical terms our problem is one of binary compound hypothesis testing. A reasonable approach to this class of problems is to employ a minimax test. Such an approach was made by Moustakides (1985) using a restricted neighborhood model for the additive noise. He found that the minimax test could be constructed as a Neyman–Pearson test between a representative pair of measures drawn from the families constituting the two hypotheses. In an attempt to generalize this work we considered using the results on minimax tests established by Huber and Strassen (1973). Huber and Strassen (H-S) prove that the minimax test between two families of measures can be reduced to a standard Neyman-Pearson test, provided that each family consists of all of the probability measures dominated (setwise) by a 2–alternating capacity. They also show that there is a sense in which this condition is necessary.

H-S theory has been successfully applied to several problems in statistical communication theory, and we were encouraged by the 'necessity' condition, and the results of Moustakides, to apply the H-S theory to the above-mentioned problem. However, we soon came to suspect that this theory was in general inapplicable to our models. We observe that the sufficient condition noted above implies that the upper envelope of each family of measures is 2–alternating. Therefore, in order to show the inapplicability of H-S theory, it suffices to prove that the upper envelope is not 2–alternating. One of the results obtained in Sadrolhefazi (1987) is

Theorem 1. Let P be a nondegenerate measure on $\boldsymbol{R}$, and let S be a subset of $\boldsymbol{R}$ that is bounded from above or from below and has more than 2 points ($|S|>2$). By a shifted version of P, denoted by T^sP we mean a probability measure Q such that for every Borel set A, we have

$$Q(A) = P(\{y \mid y=x-s \text{ for some } x \text{ in } A\}).$$

Define $\boldsymbol{P} = \{Q \mid Q = T^sP \text{ for some } s \text{ in } S\}$, and let u be its upper envelope:

$$u(A) = \sup\{Q(A) \mid Q \in \boldsymbol{P}\} \text{ for every Borel set A.}$$

If the convex hull of $\boldsymbol{P}$ consists of all the probability measures majorized by u, then u is not

2–alternating.

We note that the last assumption is a strong condition and restricts the generality that we aimed for. Therefore, for specific families of discrete and continuous measures we have developed similar results that do not resort to such assumptions. For specific probability measures such as Gaussian, exponential, uniform, we can show that it suffices to have a nontrivial shift set (i.e. a shift set with at least 2 points) to ensure that the upper envelope is not 2-alternating. The conclusions reached above continue to hold if the original family of measures is enlarged by an ε-contamination or a total variation model for the noise.

Theorem 2. Let M be the set of all probability measures on a measurable space (Ω, F). Given a nonempty subset P of M, let u_P denote its upper envelope. For any $0 < \varepsilon < 1$, define

$$C = \{ Q \in M \mid Q = (1-\varepsilon) P + \varepsilon H, P \in P, H \in M \}$$

$$V = \{ Q \in M \mid |P(A) - Q(A)| \leq \varepsilon \text{ for all A in } F, P \in P \}.$$

Then the upper envelope of C or V is 2-alternating iff u_P is.

Therefore our results apply to signal-plus-noise models that incorporate their noise uncertainty via such neighborhoods. We finally report the only nontrivial cases that we have encountered where the upper envelope *is* 2–alternating. For such simple cases, however, we do not need to use H-S theory to find a minimax test.

The success of H-S theory in robust statistics, coupled with the 'necessity' condition, can be misinterpreted as an argument for its universality. Our results, however, combined with those of Moustakides, reveal otherwise. Moustakides was able to reduce the minimax test to a Neyman–Pearson test, while we have shown that this cannot be done using any H-S type results. This indicates that the necessity condition provided by Huber and Strassen needs to be read with care.

References

[1] G. V. Moustakides, "Robust detection of signals: a large deviations approach," IEEE Trans. Info. Theory, Vol. It-31, pp. 822-825, 1985.

[2] P. J. Huber and V. Strassen, "Minimax tests and the Neyman-Pearson Lemma for capacities," Ann. Statist., Vol. 1, pp. 251-263, 1973.

[3] A. Sadrolhefazi, "Some results on the role of 2-alternating upper probabilities in signal-plus-noise models," MS Thesis, Cornell University, 1987.

BOUNDS ON THE VARIANCES OF IMPORTANCE SAMPLING SIMULATIONS IN DIGITAL COMMUNICATION SYSTEMS

D. Lu and K. Yao
Electrical Engineering Department, University of California
Los Angeles, CA 90024-1594

ABSTRACT

Bit error rate (BER) of most practical complex digital communication systems can not be evaluated by pure analytical means but must be based on simulation techniques. Various forms of importance sampling (IS) simulation approach have been proposed in order to reduce the costly large number of simulation runs. The estimation variance provides a good measure of the efficiency of one simulation approach to another. However, the evaluation of the variance and improvement ratio may be as complex as the evaluation of the BER of the system. In this paper, we propose a new and simple technique for the evaluation and bounding of the simulation estimation variance and improvement ratio. We apply this technique to the CIS and IIS simulation approaches. Various numerical results are presented to show the tightness and usefulness of this technique under many general conditions.

I. INTRODUCTION

Simulation based on the Monte Carlo (MC) technique is commonly used in the evaluation of the bit error rate (BER) of many practical complex digital communications with various random disturbances and nonlinearities. However, in order to consider high performance systems and thus requiring the estimation of low probability error events by the use of direct MC technique, very large number of MC runs are needed inevitably and can be computationally costly [1-2]. Thus, various importance sampling (IS) schemes have been proposed to make the low probability events occur more frequently. Many of these conventional importance sampling (CIS) techniques [1-2] are based on optimized variance increasing schemes operating on the original pdf's, while other more recently proposed scheme denoted as improved importance sampling (IIS) technique [3] is based on optimized translations of these pdf's. The advantages of the CIS technique over that of the MC scheme as well as that of the IIS over that of the CIS are quite significant and have been demonstrated in [1-3].

The basic problem in the use of any IS technique is to determine the number of simulation runs required for some accuracy in the estimation of the BER. By analyzing and evaluating the variance or the improvement ratio of an IS technique with respect to the MC approach, we can obtain the required number of simulation runs in some simple cases. However, for most practical digital communication systems, the analysis and evaluation of these variances and improvement ratios are quite difficult. Indeed, the complexity of the evaluation of these quantities may be as complex as the evaluation of the BER of the system. Hence, we need to find other more basic approaches to estimate these expressions. Previously, a bound on the estimation of the variance of the conventional IS approach for an one dimensional pdf has been considered [4]. However, for practical complex digital communication systems, we need to find bounds applicable to more general classes of IS simulation techniques involving multidimensional pdf's.

In Section 2, we consider some basic definitions and assumptions in the formulation of the simulation problem. In Section 3, by using a simple and basic property of the weighting function in the IS estimator, we obtain two theorems dealing with the upper bound on the estimation variance of any IS estimator and the associataed lower bound on the improvement ratio. Finally, in Section 4, we give some specific numerical results that show the tightness and usefulness of this new bounding technique for the prediction of system performances by IS simulations.

II. DEFINITIONS AND ASSUMPTIONS

Consider a digital communication system modeled by

$$Y_k = g(\underline{X}) \ , \tag{2.1}$$

where Y_k is the output sample sequence, g(.) is the system response, and $\underline{X}$ is the input random vector defined by

$$\underline{X} = [X_k, X_{k-1}, \ldots, X_{k-M+1}] = \underline{A} + \underline{N}, \tag{2.2a}$$

$$\underline{A} = [A_k, A_{k-1}, \ldots, A_{k-M+1}], \tag{2.2b}$$

$$\underline{N} = [N_k, N_{k-1}, \ldots, N_{k-M+1}], \tag{2.2c}$$

where $\underline{A}$ is the input data vector and $\underline{N}$ is an independent identically distributed noise vector.

In a binary detection problem with a decision threshold of T, the conditional BER under H_o can be given by

$$P_0 = \int_{-\infty}^{\infty} h(y-T) f^o(y) dy, \tag{2.3}$$

where $f^o(y)$ is the pdf of Y_k under H_o and h(.) is the indicator function. Then (2.3) is equivalent to

$$P_0 = \int_{-\infty}^{\infty} h[g(\underline{a}+\underline{n})-T] f_o(\underline{a}) f_1(\underline{n}) d\underline{a} d\underline{n}, \tag{2.4}$$

where $f_o(\underline{a})$ and $f_1(\underline{n})$ are the pdfs of the input data and noise vectors, respectively.

In any IS scheme, we modify the original pdf $f_1(\underline{n})$ to a new pdf $f_1^*(\underline{n})$. The weighting function can be defined by

$$w(\underline{n}) = f_1(\underline{n})/f_1^*(\underline{n}). \tag{2.5}$$

Then (2.4) can be reexpressed as

$$P_0 = \int_{-\infty}^{\infty} h[g(\underline{a}+\underline{n})-T]w(\underline{n})f_1^*(\underline{n})f_o(\underline{a})d\underline{a}d\underline{n} \tag{2.6a}$$

$$= \int_{\Omega} w(\underline{n})f_1^*(\underline{n})f_o(\underline{a})d\underline{a}d\underline{n} \tag{2.6b}$$

where

$$\Omega = \{ (\underline{a},\underline{n}): g(\underline{a}+\underline{n}) \geq T \}. \tag{2.7}$$

From (2.6a), an IS estimator for the p_0 is given by

$$P_0^* = (1/N^*) \sum_{j=1}^{N^*} h[g(\underline{a}+\underline{n})]w(\underline{n}) , \tag{2.8}$$

where N^* is the simulation runs required in the IS approach.

It is simple to show that the IS estimator is unbiased and the estimation variance is given by

$$\sigma_{IS}^2 = (1/N^*) \int_{-\infty}^{\infty} h[g(\underline{a}+\underline{n})-T][w(\underline{n})-P_o]f_1(\underline{n})f_o(\underline{a})d\underline{a}d\underline{n}$$

$$= (1/N^*) \int_{\Omega} [w(\underline{n})-P_o]f_1(\underline{n})f_o(\underline{a})d\underline{a}d\underline{n}. \tag{2.9}$$

By letting $w(\underline{n})=1$, the MC estimation variance is given by

$$\sigma_{MC}^2=(1/N) \int_{-\infty}^{\infty} h[g(\underline{a}+\underline{n})-T][1-P_o]f_1(\underline{n})f_o(\underline{a})d\underline{a}d\underline{n}. \tag{2.10}$$

The improvement ratio of the IS to the MC simulations can be defined by

$$r_{MC/IS} = N\sigma^2_{MC}/N^*\sigma^2_{IS} . \tag{2.11}$$

From (2.11), the $r_{MC/IS}$ expression describes the reduction of simulation runs of the IS approach with respect to the direct MC scheme under the same estimation variance as well as the reduction of the IS estimation variance under the same number of runs.

III. BOUNDS ON THE ACCURACY OF IS ESTIMATION AND IMPROVEMENT RATIO

Two simple but basic theorems dealing with bounds on the accuracy of IS estimation and improvement ratio are given below.

Theorem 1.
The estimation variance in (2.9) is upper bounded by

$$\sigma^2_{IS} \leq [P_o/N^*][\operatorname{Sup}_{\underline{n}} w(\underline{n}) - P_o] . \tag{3.1}$$

Theorem 2.
The improvement ratio $r_{MC/IC}$ of (2.11) is lower bounded by

$$r_{MC/IS} \geq [1 - P_o]/[\operatorname{Sup}_{\underline{n}} w(\underline{n}) - P_o] . \tag{3.2}$$

IV. APPLICATION OF BOUNDING TECHNIQUE

Now, we consider some examples to illustrate the usefulness and tightness of these bounds.

1. Linear System with no Memory

First, consider a simple linear memoryless system where under H_o the signal is given by $-A$. thene the decision region can be expressed as

$$\Omega = \{n : n - A \geq 0\} , \tag{4.1}$$

and the noise is Gaussian with a pdf given by

$$f_1(n) = (2\pi\sigma^2)^{-.5}\exp(-n^2/2\sigma^2) , \quad -\infty < n < \infty . \tag{4.2}$$

In the CIS approach, the original pdf $f_1(n)$ is modified to a new pdf $f_{1*}(n)$ given by

$$f_1^*(n) = (2\pi\sigma_*^2)^{-.5}\exp(-n^2/2\sigma_*^2) , \quad -\infty < n < \infty , \tag{4.3}$$

which is still Gaussian with a new variance $\sigma_*^2 \geq \sigma^2$. The weighting function is then given by

$$w_{CIS}(n) = (\sigma_*/\sigma)\exp\{-(1-\sigma^2/\sigma_*^2)n^2/2\sigma^2_*\} . \tag{4.4}$$

From [3], the optimum value of σ_* is given by

$$\sigma_*(opt) = A . \tag{4.5}$$

Then the supremum value of the weighting function is given by

$$\sup w(n) = \max w(n) = (A/\sigma)\exp[-(1-\sigma^2/A^2)A^2/2\sigma^2] \qquad (4.6)$$

In the IIS approach, we translate the original r.v. n by the amount c such that the new pdf $f_1^{**}(n)$ is given by

$$f_1^{**}(n) = f_1(n-c) \; . \qquad (4.7)$$

Then the weighting function is given by

$$w_{IIS}(n) = (2\sigma^2)^{-1} \exp[-c(2n-c)/2\sigma^2] \; . \qquad (4.8)$$

The parameter c is chosen to minimize the estimation variance [3]. For this problem, the optimum c is given by

$$c(opt) = A \; . \qquad (4.9)$$

Then the superemum weighting function value is given by

$$\sup w(n) = \max w(n) = \exp[-A^2/2\sigma^2] . \qquad (4.10)$$

At P_o = 1E-10, the signal-noise ratio $A/\sigma = 6.36$, using (4.6) and (4.10) in (3.2), we obtain

$$\sigma^2_{CIS}/P_o^2 = 172.6/N^*_{CIS} \qquad (4.11)$$

$$\sigma^2_{IIS}/P_o^2 = 16.5/N^*_{IIS} . \qquad (4.12)$$

For a relative variance (i.e., σ_{IS}^2/P_o^2) equal to 0.01, from (4.11) and (4.12), upper bounds on the required number of simulation runs for CIS and IIS cases are given by

$$N^*_{CIS} \le 17{,}250 \; , \qquad (4.13)$$

$$N^*_{IIS} \le 1{,}650 \; . \qquad (4.14)$$

Similarly, the lower bounds on the improvement ratios for CIS and IIS cases are given by

$$r_{MC/CIS} \ge 5.8E7 \; , \qquad (4.15)$$

$$r_{MC/IIS} \ge 6.0E8 \; . \qquad (4.16)$$

For practical complex digital communication systems, we need to consider more general context. For this reason, the generalized exponential distribution (GED) has been introduced to describe the random disturbances [2]. The pdf of the GED can be expressed by

$$f_v(n) = [v/2\sqrt{2}\sigma\Gamma(1/v)] \exp\{-|n/\sqrt{2}\sigma|^v\} \; , \qquad (4.17)$$

where σ is a parameter, b(.) is the gamma function and v is a parameter which can be taken to be equal to 2 for the Gaussian pdf and equal to 1 for the Laplace pdf.

For $1<v<2$, the BER cannot be obtained by pure analysis. If we want to use IS techniques to estimate the BER, the variances and the improvement ratio also cannot be evaluate by previous approaches. Fortunately, this problem can be solved by making use of our bounding technique.

In order to verify on the tightness of these bounds, corresponding exact results are tabulated in the Table 1. In the first row we consider the one-dimensional Gaussian case. As can be seen, the bounds differ from the exact values by approximately factors of 2. We now proceed to show that similar advantages are still valid for other more general cases. For the same BER and relative variance, in the last row of Table 1, we consider a one- dimensional Laplace case, while from the second to the fifth row we consider the one-dimensional generalized exponential case. This results show the advantage of the IIS approach to that of the CIS approach as well as to the usefulness of the bounding technique. Places in the columns with an x represent results that can not be evaluated by analytical means but bounding results are available.

2. Linear System with Memory

Suppose the output of the system response is modeled as a linear summation, then (2.1) becomes

$$Y_k = \sum_{i=k}^{k-M+1} b_{k-i} \, [a_i(j) + n(i)] \ . \tag{4.18}$$

where $\{b_i\}$ is the real-valued system impulse-response sequence, $a_i(j)$ is the i-th sample of the j-th signal realization $\underline{a}(j)$ and $n(i)$ is the i-th sample of input noise under $a_i(j)$.

The decision rule is given by

$$\sum_i b_{k-i} \, [a_i(j) + n(i)] \geq T \ . \tag{4.19}$$

By letting $T_j = T - \sum_i b_{k-i} \, a_i(j)$,

we have

$$b_{k-i} \, n(i) \geq T_j \ . \tag{4.20}$$

In the CIS approach, under the signal realization $\underline{a}(j)$, the weighting function can be written as [3]

$$w_{CIS}(\underline{n}) = (\sigma_*/\sigma)^M \exp[-(1-\sigma^2/\sigma^2_*) \sum_i n^2(i)/2\sigma^2] \ . \tag{4.21}$$

It is clear that the w_{CIS} is a monotone decreasing function on $\sum_i n^2(i)$, so that

$$\sup_{\underline{n}} w_{CIS}(\underline{n}) = (\sigma_*/\sigma)^M \exp[-(1-\sigma^2/\sigma^2_*) \inf_{\underline{n}} \sum_i n^2(i)/2\sigma^2]. \tag{4.22}$$

In order to find $\inf \sum_i n^2(i)$ subject to the constraint (4.20), the method of Lagrange multiplier is employed. The result is given by

$$\inf_{\underline{n}} \sum_i n^2(i) = \min_{\underline{n}} \sum_i n^2(i) = T_j^2/\beta^2 , \tag{4.23}$$

where

$$\beta^2 = \sum_{i=k}^{k-M+1} b^2_{k-i} .$$

Substituting (4.23) into (4.22) gives

$$\sup_{\underline{n}} w_{CIS}(\underline{n}) = (\sigma_*/\sigma)^M \exp[-(1-\sigma^2/\sigma^2_*) T_j^2/2\sigma^2\beta^2] . \tag{4.24}$$

Similarly, for IIS approach we have

$$\sup_{\underline{n}} w_{IIS}(\underline{n}) = \exp(-T_j^2/2\sigma^2\beta^2) . \tag{4.25}$$

For a system with ISI of memory $M=3$ and $b_{k-i}=0.5^{k-i}$, inserting (4.24)-(4.25) into (3.4), the improvement ratios can be obtained, and are given in Table 2. As can be seen, these bounds are tight as compared to the exact values for the multi-dimensional Gaussian case.

V. CONCLUSIONS

We have presented a new and useful approach to the estimation variance of any IS simulation. Expressions and bounds applicable to the evaluation of simulation runs in order to meet certain accuracy requirements as well as improvement ratios have also been given. Numerical results verify the tightness and usefulnes of this approach particularly for these problems that do not have obvious analytical solutions. We believe this approach is of practical use in simulation applications for many complex digital communication systems.

VI. ACKNOWLEDGEMENT

This work at UCLA is partially supported under NASA contract NAS 2-304.

REFERENCES

[1] K. S. Shanmugun and P. Balaban, 'A Modified Monte Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication', IEEE Trans. Commu. vol. COM-28, pp.1916-1924, Nov. 1980.

[2] M. C. Jeruchim, 'Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication System', IEEE Journal, vol. SAC-2, No.1, pp.153-170, Jan. 1984.

[3] D. Lu and K. Yao, 'Improved Importance Sampling Technique for Efficient Simulation of Digital Communication Systems', to appear in IEEE Journal on Selected Areas in Communications, Jan. 1988.

[4] G. W. Lank, 'Theoretical Aspects of Importance Sampling Applied to False Alarms', IEEE Trans. Imform. Theory, vol. IT-29, No.1, pp.73-82, Jan. 1983.

Table 1. Simulation Runs and Improvement Ratios from Bounds and Analysis for P_E=1E-10, σ^2_{IS}/P_o^2=.01 in the one-dimensional case

Distribution		Runs		Improvements Ratios	
		Bounds	Exact	Bounds	Exact
C I S					
Gaussian		1.72E4	8.3E3	5.80E7	1.20E8
Generalized Exponential	v=1.8	1.36E4	x	7.37E7	x
	v=1.6	1.29E4	x	7.75E7	x
	v=1.4	1.28E4	x	7.78E7	x
	v=1.2	1.28E4	x	7.79E7	x
Laplace		1.20E4	6.1E3	8.30E7	1.64E8
I I S					
Gaussian		1650	700	0.60E9	1.42E9
Generalized Exponential	v=1.8	951	x	1.05E9	x
	v=1.6	677	x	1.48E9	x
	v=1.4	460	x	2.18E9	x
	v=1.2	271	x	3.68E9	x
Laplace		100	100	1.00E10	1.0E10

Table 2. Improvement Ratios $\gamma_{MC/IS}$ for A Digital Communication System with ISI of Memory M=3

P_0	A/σ	$\gamma_{MC/CIS}$		$\gamma_{MC/IIS}$	
		Bounds	Exact	Bounds	Exact
1E-3	12.14	2.0	6.4	3.5E1	1.1E2
1E-4	15.44	8.6	2.8E1	3.0E2	7.8E2
1E-5	18.15	4.7E1	1.6E2	2.4E3	6.3E3
1E-6	20.48	2.8E2	1.0E3	2.2E4	5.4E4
1E-7	22.64	1.9E3	6.9E3	2.0E5	4.8E5
1E-8	24.61	1.4E4	5.4E4	1.9E6	4.3E6

ON THE APPLICATION OF IMPORTANCE SAMPLING TO THE ANALYSIS OF DETECTION SYSTEMS

GEOFFREY ORSAK AND BEHNAAM AAZHANG

Department of Electrical and Computer Engineering
Rice University
Houston, TX 77251-1892

ABSTRACT

Analytical solutions of the performance of nonlinear communication systems have been very difficult to obtain. Monte Carlo simulations are often the most tractable method of estimating P_e. A technique known as Importance Sampling can substantially reduce the number of simulation trials. We give here a general method of applying this technique to the optimal detection problem.

I. INTRODUCTION

In recent years, much research has centered on nonlinear communication systems. Analytical solutions of the performance of these systems have been very difficult to obtain. Therefore efforts have been concentrated on obtaining asymptotic approximations and bounds for estimating system error probability [1, 7, 14, 15]. The *Asymptotic Relative Efficiency* (ARE) and and the *Central Limit Theorem* (CLT) have been the two most powerful tools for simplifying the performance analysis of the nonlinear system. Both of these require infinitely large sample sizes for analytically exact solutions. This is not the case in a realistic communication system. Therefore, the use of the ARE and CLT yield estimates of system performance. The accuracy of these estimates have been investigated for finite sample sizes. Results showed that for certain distributions, the use of the ARE as a performance measure could be misleading [9].

Another, often more tractable method for obtaining performance estimates is Monte Carlo simulations. This method can produce very accurate estimates of system performance if the number of simulation trials is sufficiently large. The complication today is that many present optimum or nearly optimum nonlinear systems operate with a very low P_e. This puts a tremendous computational burden on the computer system. As an example, for a 95% confidence interval of $\left[2P_e/5\,,8P_e/5\right]$, there is a requirement of approximately $10/P_e$ simulation trials [16]. This can be quite large for $P_e < 10^{-7}$, thus making this method presently impractical.

A technique known as *Importance Sampling* has been employed to greatly reduce the number of simulation trials [2, 5, 6, 12]. This technique is based on biasing the noise distribution such that more samples are taken from the *important* regions (regions that cause errors) and then unbiasing the estimate of system performance. Within certain restrictions, it can be shown that this biasing technique will reduce the variance of the estimate, thereby reducing the number of simulation trials for the same level of accuracy. Work by Shanmugam and Balaban has demonstrated the effectiveness of Importance Sampling to the communication system problem, but there has been no general technique developed for utilizing this powerful method. Spaulding [13] attempted to apply Importance Sampling to the

---

This work supported by the National Science Foundation under contract NCR-8710844.

analysis of LOBD systems, but found no good method of biasing the noise and therefore used standard Monte Carlo simulations. Lank [8] examined a specific type of biasing (scaling), but most of the previous work has been limited to Gaussian noise in a particular system [3, 4]. In this work, we describe an alternate method of biasing the noise for a reduced variance estimate of P_e which will apply to a large class of detection systems and noise density functions.

II. PROBLEM FORMULATION

In the classical detection problem, we wish to determine between two hypotheses:

$$\begin{aligned} H_0 &: \underline{R} = \underline{\eta} \\ H_1 &: \underline{R} = \underline{S} + \underline{\eta} \end{aligned} \tag{1}$$

where $\underline{R}$ is the N-dimensional received vector, $\underline{S}$ is the N-dimensional signal vector, and η is additive noise. Define $R^N \subset \mathbb{R}^N$ to be the region that H_0 is chosen and $\overline{R}^N = R^{N^c} \subset \mathbb{R}^N$ to be the region that H_1 is chosen. The average probability of a wrong decision for a symmetric detector can be written as

$$P_e = \int_{\mathbb{R}^N} I(\underline{R})\, f_{\underline{R}}(\underline{R}|\, H_0)\, d\underline{R} \tag{2}$$

where $f_{\underline{R}}(\underline{R}|\, H_0)$ is the density function of the received vector conditioned on the hypothesis H_0 and $I(\underline{R})$ is the indicator function over $\overline{R}^N$. The Maximum Likelihood Estimator for P_e using the binomial model is

$$\hat{P}_e = \frac{1}{M}\sum_{i=1}^{M} I(\underline{R}_i) \tag{3}$$

where M is the number of trials of the simulation and $\underline{R}_i$ is the sample vector with variance

$$\operatorname{var}\left[\hat{P}_e\right] = \frac{P_e(1-P_e)}{M}\ . \tag{4}$$

This section has described the Monte Carlo method for estimating P_e. The variance of this estimate for $P_e << \frac{1}{2}$ is approximately $\frac{P_e}{M}$. The following sections will demonstrate a method to substantially reduce the variance of this estimate.

III. IMPORTANCE SAMPLING

In this section, Importance Sampling is introduced. This technique reduces the variance of the estimate of P_e by generating data $\underline{R}$ from the biased density function $f_{\underline{R}^*}(\underline{R}|\, H_0)$. This joint density is chosen such that the random vector $\underline{R}^*$ is more likely to come from the important regions in $\mathbb{R}^N$. This biasing will enable us to more accurately estimate P_e. Since $f_{\underline{R}}(\underline{R}|\, H_0)$ and $f_{\underline{R}^*}(\underline{R}|\, H_0)$ are the joint density functions of the random noise vector for $\underline{R}$ and $\underline{R}^*$, we will refer to them as $f_{\underline{R}}(\underline{R})$ and $f_{\underline{R}^*}(\underline{R})$ for the duration of this paper. The resulting estimator for P_e using $f_{\underline{R}^*}(\underline{R})$ is given as

$$P_e^* = \frac{1}{M}\sum_{i=1}^{M} W(\underline{R}_i)\, I(\underline{R}_i) \tag{5}$$

where $I(\underline{R})$ is as before the indicator function over $\overline{R}^N$, $\underline{R}_i$ is the i^{th} sample vector from density $f_{\underline{R}^*}(\underline{R})$, and $W(\underline{R}_i)$ is an error weighting function designed so that P_e^* be an unbiased estimator, i.e., $E\left[P_e^*\right] = P_e$. This equality will hold when

$$W(\underline{R}) = \frac{f_{\underline{R}}(\underline{R})}{f_{\underline{R}^*}(\underline{R})} \, . \tag{6}$$

Using this weight function in (5) results in an unbiased estimate of P_e for the communication system with noise density $f_{\underline{R}}(\underline{R})$. Note that the estimator P_e^* still counts errors based upon the original decision rule of choosing H_0 if $\underline{R} \in \mathcal{R}^N$ and choosing H_1 if $\underline{R} \in \overline{\mathcal{R}}^N$, but each error is now weighted by $W(R)$.

The variance of P_e^* can be shown to be

$$\mathrm{var}_{f^*}\left[P_e^*\right] = \frac{\overline{W} - P_e^2}{M} \quad \text{where } \overline{W} = \int_{\overline{\mathcal{R}}^N} \frac{f_{\underline{R}}(\underline{R})}{f_{\underline{R}^*}(\underline{R})} f_{\underline{R}}(\underline{R})\, d\underline{R} \, . \tag{7}$$

Therefore, we have a method of estimating P_e which uses the original decision rule but input density function $f_{\underline{R}^*}(\underline{R})$ rather than $f_{\underline{R}}(\underline{R})$. The estimate is then unbiased by scaling the errors with the weighting function $W(\underline{R})$. We must note that the variance of P_e^* is dependent upon $\overline{W}$, P_e, and M, whereas the variance of $\hat{P}_e$ is dependent upon only P_e and M (see (4)). The parameter $\overline{W}$ will allow for the reduction of the variance of the estimate for a fixed M and a given system.

IV. SELECTION OF $f_{\underline{R}^*}(\underline{R})$

The problem now is to determine $f_{\underline{R}^*}(\underline{R})$ such that $\mathrm{var}_{f^*}\left[P_e^*\right]$ is minimized for all values of P_e. From (7), we see that $\overline{W}$ must be minimized with respect to $f_{\underline{R}^*}(\underline{R})$ since $\overline{\mathcal{R}}^N$ and $\mathcal{R}^N$ are determined by the form of the detector.

The optimal solution is found by the Calculus of Variations to be

$$f_{\underline{R}^*_{opt}}(\underline{R}) = \frac{I(\underline{R})\, f_{\underline{R}}(\underline{R})}{\int_{\overline{\mathcal{R}}^N} I(\underline{R})\, f_{\underline{R}}(\underline{R})\, d\underline{R}} = \frac{I(\underline{R})\, f_{\underline{R}}(\underline{R})}{P_e} \, . \tag{8}$$

It is clear that this is a degenerative solution to the optimization problem because we must know the value we are trying to estimate. This density function always generates errors by restricting the random vector to be an element of $\overline{\mathcal{R}}^N$ and then weights each error by P_e, thereby giving a zero variance estimate.

This shows that there exists an $f_{\underline{R}^*_{opt}}(\underline{R})$ that enables us to have a zero variance estimate, but to generate data with this density function we must know P_e. Clearly, simulations are not required in this situation. We must then search for a suboptimal $f_{\underline{R}^*}(\underline{R})$ that yields a reduced variance estimate.

For Importance Sampling to be a useful method for estimating P_e, it is necessary that $\mathrm{var}\left[P_e^*\right] \le \mathrm{var}\left[\hat{P}_e\right]$ or equivalently $\overline{W} \le P_e$. Conditions for this to hold are easily recognized by comparing the integral forms of $\overline{W}$ and P_e. We wish that

$$\overline{W} = \int_{\overline{\mathcal{R}}^N} \frac{f_{\underline{R}}(\underline{R})}{f_{\underline{R}^*}(\underline{R})} f_{\underline{R}}(\underline{R})\, d\underline{R} \le \int_{\overline{\mathcal{R}}^N} f_{\underline{R}}(\underline{R})\, d\underline{R} = P_e \, . \tag{9}$$

From this it is clear that

$$f_{\underline{R}^*}(\underline{R}) \ge f_{\underline{R}}(\underline{R}) \quad \forall \ \underline{R} \in \overline{\mathcal{R}}^N \tag{10}$$

is a sufficient condition for $\overline{W} \le P_e$. Since the regions $\overline{\mathcal{R}}^N$ and $\mathcal{R}^N$ are not readily available, we must develop some technique where the choice of $f_{\underline{R}^*}(\underline{R})$ can be made without explicit knowledge of $\overline{\mathcal{R}}^N$. To do this, we need the following theorem.

Theorem : Let $\bar{R}^N$ be the minimum P_e decision region for H_0 in the hypotheses testing problem stated in (1). Define a new signal $\hat{\underline{S}} = \alpha \underline{S}$ where $0 \le \alpha \le 1$. Let Z_0 to be the minimum P_e decision region for H_0 with signal $\hat{\underline{S}}$. Then under the assumption of i.i.d. data and some mild conditions on the noise density $f_R(\underline{R})$, we have $Z_0 \supseteq \bar{R}^N$.

To prove this theorem, we will need to on the maximum likelihood decision rule. The minimum P_e decision rule for i.i.d. data is

$$L(\underline{R}) = \prod_{j=1}^{N} \frac{f_R(R_j - S_j)}{f_R(R_j)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 , \tag{11}$$

where $L(\underline{R})$ is the likelihood ratio. Note that $f_R(R)$, the marginal density of the noise, is equal to $f_{R_j}(R)$ for all j since we assumed i.i.d. noise. Since $f_R(0)$ is not equal to zero, we can rewrite the decision rule as

$$-\sum_{j=1}^{N} \ln\left[f_R(R_j - S_j)/f_R(0)\right] \underset{H_1}{\overset{H_0}{\gtrless}} -\sum_{j=1}^{N} \ln\left[f_R(R_j)/f_R(0)\right] . \tag{12}$$

Defining a function based on the likelihood ratio in (11) as

$$D(\underline{X} - \underline{Y}) \triangleq -\sum_{j=1}^{N} \ln\left[f_R(X_j - Y_j)/f_R(0)\right] , \tag{13}$$

we can state the decision rule as

$$D(\underline{R} - \underline{S}) \underset{H_1}{\overset{H_0}{\gtrless}} D(\underline{R} - \underline{0}) . \tag{14}$$

In order to prove the theorem we will need to constrain the marginal density $f_R(X)$ so that $D(\underline{X})$ is a metric with the additional property that $D(\alpha\underline{X}) \le \alpha D(\underline{X})$ for $0 \le \alpha \le 1$. We start by placing some appropriate conditions on the marginal density of the noise that it is both symmetric and unimodal, i.e.,

$$\begin{aligned} &f_R(X) = f_R(-X) \quad \text{and} \\ &f_R(R) > f_R(X) \quad \forall \ |X| > |R| . \end{aligned} \tag{15}$$

Assuming (32), it is easily verified that $D(\underline{X})$ has the following properties:

$$\begin{aligned} &\text{(a) } D(\underline{X}) \ge 0 \quad \forall\, \underline{X} \\ &\text{(b) } D(\underline{X}) = D(-\underline{X}) \\ &\text{(c) } D(\underline{X} - \underline{Y}) = 0 \quad \text{iff } \underline{X} = \underline{Y} . \end{aligned} \tag{16}$$

It is easy to show that the decision rule in (11) can be stated as in (14) where $D(\underline{X})$ is a metric with the additional property (e) under the following conditions on the marginal density of the noise,

$$\begin{aligned} &(1)\ f_R(X) = f_R(-X) \\ &(2)\ f_R(X) > f_R(Y) \quad \forall \ |Y| > |X| \\ &(3)\ f_R(X) f_R(Y) \le f_R(0) f_R(X+Y) \\ &(4)\ \frac{f_R(0)}{f_R(\alpha X)} \le \left[\frac{f_R(0)}{f_R(X)}\right]^{\alpha} \quad 0 \le \alpha \le 1 . \end{aligned} \tag{17}$$

Assuming that conditions in (17) hold for the detection problem in (11) we are prepared to prove the theorem.

Proof: The region that H_0 is chosen can be expressed in terms of the metric $D(\underline{X})$ as $\bar{R}^N = \{\underline{R} : D(\underline{R}-\underline{S}) \le D(\underline{R})\}$. Similarly write Z_0 as $Z_0 = \{\underline{R} : D(\underline{R}-\alpha\underline{S}) \le D(\underline{R})\}$. We now show that $Z_0 \supseteq \bar{R}^N$ or equivalently that if $\underline{R} \in \bar{R}^N$, then $\underline{R} \in Z_0$. In other words, we wish to show that if $\underline{R} \in \bar{R}^N$, then $D(\underline{R}-\alpha\underline{S}) \le D(\underline{R})$.

We start by writting $D(\underline{R}-\alpha\underline{S})$ as

$$D(\underline{R}-\alpha\underline{S}) = D((1-\alpha)\underline{R} + \alpha\underline{R} - \alpha\underline{S}).$$

By the triangle inequality

$$D(\underline{R}-\alpha\underline{S}) \le D((1-\alpha)\underline{R}) + D(\alpha(\underline{R}-\underline{S}))$$

Since we assumed property (e) holds for $D(\underline{X})$ and that $\underline{R} \in \bar{R}^N$ we can write

$$D(\underline{R}-\alpha\underline{S}) \le (1-\alpha)D(\underline{R}) + \alpha D(\underline{R}-\underline{S}) \le (1-\alpha)D(\underline{R}) + \alpha D(\underline{R}) \le D(\underline{R}) \ \blacksquare$$

We now will apply this theorem to the problem of selecting $f_{\underline{R}^*}(\underline{R})$ such that the sufficient condition in (10) for a reduced variance estimate of P_e is satisfied. From the theorem we see that

$$Z_0 = \left\{\underline{R} : \frac{f_{\underline{R}}(\underline{R}-\alpha\underline{S})}{f_{\underline{R}}(\underline{R})} \ge 1\right\} \supseteq \left\{\underline{R} : \frac{f_{\underline{R}}(\underline{R}-\underline{S})}{f_{\underline{R}}(\underline{R})} \ge 1\right\} = \bar{R}^N \tag{18}$$

for density functions with the properties stated in (17). Therefore, selecting

$$f_{\underline{R}^*}(\underline{R}) = f_{\underline{R}}(\underline{R}-\alpha\underline{S}) \ge f_{\underline{R}}(\underline{R}) \quad \forall\ \underline{R} \in \bar{R}^N \text{ and } 0 \le \alpha \le 1 \tag{19}$$

fulfills the sufficient condition in (10) for a reduced variance estimate of P_e. The choice of $f_{\underline{R}^*}(\underline{R})$ is then simply a shifted version of the original noise density, i.e., the random vector $\underline{R}^* = \underline{R} + \alpha\underline{S}$. This allows for easy implementation in computer simulations.

To determine how well this choice of $f_{\underline{R}^*}(\underline{R})$ works for estimating P_e using Importance Sampling, we will need some criterion which can be used to compare it to the standard Monte Carlo estimate. Since Importance Sampling is generally employed to reduce the number of simulation trials , we choose to compare the number of trials required for the same estimate variance using the two methods. More exactly, we will use the performance measure [12]

$$\gamma \overset{\Delta}{=} \frac{M_{MC}}{M_{IS}} = \frac{P_e - P_e^2}{\overline{W} - P_e^2}, \tag{20}$$

where M_{MC} is the number of samples required for a fixed variance using standard Monte Carlo simulations and M_{IS} is the number of samples required for the same variance using Importance Sampling.

Clearly we would like for γ to be as large as possible, particularly for small P_e. We, therefore, would like to choose α in (19) so that γ is maximized. This choice of α is specific to every noise density and SNR, but $\alpha = 1/2$ will be seen to be quite robust for small P_e, thus we have the following proposition for the choice of $\alpha = \frac{1}{2}$.

Proposition 1: When estimating P_e for a one dimensional vector $\underline{R}$ with a non-zero, symmetric noise density $f_{\underline{R}}(\underline{R})$, the $\lim_{P_e \to 0} \gamma = \infty$ for the choice of $f_{\underline{R}^*}(\underline{R}) = f_{\underline{R}}(R - S/2)$ [11].

This Proposition shows that at least for the one dimensional case , $\alpha = 1/2$ is asymptotically optimum in the sense that the gain in using Importance Sampling over Monte Carlo simulations increases without bound as P_e tends to zero. In summary, we should state that direct minimization of the var $\left[P_e^*\right]$ resulted in an unrealistic choice since it required the knowledge of P_e. By the use of the theorem, we were able to choose a suboptimal $f_{\underline{R}^*}(\underline{R})$ such that a reduced variance estimate of P_e was obtained for a class of optimal detection systems. Finally, Proposition 1. showed that asymptotically, the choice of $\alpha = 1/2$ for a one dimensional system, will dramatically reduce the number of simulation trials for a fixed variance over the standard Monte Carlo method for any symmetric marginal noise density with infinite support. With the understanding of the theory of Importance Sampling, we now apply it to a few detection problems.

V. EXAMPLES

For this method of biasing the input noise density $f_{\underline{R}}(\underline{R})$ to be useful, we must show that the restrictions on the marginal density in (17) allow for a wide class of density functions to be simulated using Importance Sampling. We will not attempt to describe here all the possible density functions which satisfy (17), but will show only one very useful, and much examined family, the generalized Gaussian density function. This family has been used as a noise model in the design of many communication systems [10]. We will calculate α_{opt} and the corresponding γ for the most common densities of this family. We start then with the following proposition.

Proposition 2: The generalized Gaussian density function $(p \geq 1)$ satisfies the conditions in (17). [11]

This proposition guarantees that shifting the noise by $\alpha\underline{S}$ will reduce the number of samples for a fixed estimator variance in optimal detection systems designed using a generalized Gaussian noise model. The question now is to determine α such that this variance is minimized.

To solve for α_{opt} and the resulting γ, we must derive an expression for $\overline{W}$. From (18), we see that we must know $\overline{R}^N$, which is generally difficult to obtain. Therefore, we can determine these quantities only in simple examples. Consequently, we will derive the optimal α and the corresponding γ for two Generalized Gaussian densities, one dimensional Laplacian noise and N dimensional Gaussian noise.

One Dimensional Laplacian Noise:

In this example the noise density function is $f_R(r) = \frac{1}{\sqrt{2}\sigma} e^{-\sqrt{2}|r|/\sigma}$. This density results in the following form for $\overline{W}$,

$$\overline{W} = \frac{1}{\sqrt{2}\sigma} \int_{S/2}^{\infty} \frac{e^{-2\sqrt{2}|r|/\sigma}}{e^{-\sqrt{2}|r-\alpha S|/\sigma}} \, dr \ . \tag{21}$$

Minimizing with respect to α yields

$$\alpha^{opt} = \begin{cases} \dfrac{\ln(4)\sigma}{3\sqrt{2}S} + \dfrac{1}{2} & \text{for } \dfrac{\sigma^2}{S^2} \leq \dfrac{9}{2\ln^2(4)} \\ 1 & \text{for } \dfrac{\sigma^2}{S^2} > \dfrac{9}{2\ln^2(4)} \end{cases} \tag{22}$$

with the corresponding $\overline{W}_{\min} = .396\, e^{-\sqrt{2}\frac{S}{\sigma}}$.

In examples with large values of SNR, $\alpha_{\text{opt}} \approx 1/2$. Calculating γ for Laplacian noise using $\overline{W}_{\min}$ yields

$$\gamma_{\text{laplacian}} = \frac{P_e(1-P_e)}{.584\, P_e^2} \approx \frac{1.71}{P_e}. \tag{23}$$

N Dimensional Gaussian Noise:

Solving for α_{opt} and γ in a similar fashion, yields

$$\alpha_{\text{opt}} = \begin{cases} \sqrt{\dfrac{1}{4} + \dfrac{\sigma^2}{N\,\overline{S}^2}} & \text{for } \dfrac{\sigma^2}{N\,\overline{S}^2} \le \dfrac{3}{4} \\ 1 & \dfrac{\sigma^2}{N\,\overline{S}^2} > \dfrac{3}{4} \end{cases} \tag{24}$$

and the corresponding γ is

$$\gamma = \frac{Q(\overline{S}\sqrt{N}/2\sigma) - Q^2(\overline{S}\sqrt{N}/2\sigma)}{\exp\left[\overline{S}^2 N/4\sigma^2 + 1\right] Q(\overline{S}\sqrt{N}/2\sigma + \sqrt{\overline{S}^2 N/4\sigma^2 + 1}) - Q^2(\overline{S}\sqrt{N}/2\sigma)} \tag{25}$$

where $\overline{S}$ is the average of S_j. This can be approximated for small P_e as

$$\gamma \approx \frac{2\sigma}{\sqrt{N}\,\overline{S}\,P_e} \tag{26}$$

See figure (1) for plots of γ versus P_e for both 10-d Gaussian and 1-d Laplacian noise. Note that γ is quite large for small P_e for both systems, demonstrating the effectiveness of this biasing technique.

γ is generally the most important quantity when measuring the usefulness of Importance Sampling since it indicates the reduction in the number of simulation trials. However in practice it is also useful to know how many trials are needed for a particular estimate accuracy. This accuracy is often expressed in terms of a confidence interval for P_e.

Recall that in Monte Carlo simulations, $10/P_e$ trials is required for a 95% confidence interval of $\left[2\hat{P}_e/5, 8\hat{P}_e/5\right]$. We can use γ to find the equivalent number of trials needed in estimating P_e using Importance Sampling. This yields

$$M_{IS} = \frac{M_{MC}}{\gamma} = \frac{10}{P_e\gamma}. \tag{27}$$

For N dimensional Gaussian noise, using γ in (26) yields $M_{IS} = 5\sqrt{N}\frac{\overline{S}}{\sigma}$ and for one dimensional Laplacian noise using γ in (23), the required number of trials to achieve the above 95% confidence interval is $M_{IS} = 5.88$. Note that for Laplacian noise, M_{IS} is independent of P_e. See figure (2) for plots of M_{IS} versus P_e for the above confidence interval for ten dimensional Gaussian noise and one dimensional Laplacian noise. Note that the maximum number of simulation trials is 100 for $P_e > 10^{-20}$.

There are situations when we can not analytically evaluate γ, but can obtain a sample estimate γ^* using I.S. To demonstrate the validity of this estimator, see figure(3) for a comparison of γ and γ^* as a function of error probability for one dimensional Laplacian noise using 100 samples per estimate. From this figure, we can see that γ^* is a quite accurate estimate of γ.

With this ability to estimate γ, we are better able to test the usefulness of biasing the noise by shifting the original density. We can now evaluate estimates of gain using Importance Sampling for more complicated (higher order) systems. Since in these cases, we may not be able to find analytical expressions for $\overline{W}$ and subsequently solve for α_{opt}, we will need to use a robust value for α.

Recall from Proposition 1. that the choice $\alpha = 1/2$ resulted in an asymptotically ideal gain. Moreover, in the two previous examples, the $\lim_{P_e \to 0} \alpha_{opt} = 1/2$. This seems to imply that the choice $\alpha = 1/2$ is somewhat robust. We will therefore use this value in the following examples.

We first examine higher order detectors with Laplacian noise. Figure (4) contains plots of γ^* versus P_e^* for one, five and ten dimensional optimal systems. Note that the gain decreases with an increase in system order, but there is still a substantial improvement over the standard Monte Carlo method. This, combined with the previous results for γ depicted in figure (1) shows that choosing $f_{\underline{R}^*}(\underline{R})$ as an appropriately shifted version of $f_{\underline{R}}(\underline{R})$ will dramatically reduce the number of simulation trials for systems having marginal densities satisfying (17), but in situations where $f_R(R)$ doesn't satisfy (17), choosing $f_{\underline{R}^*}(\underline{R}) = f_{\underline{R}}(\underline{R} - 1/2\underline{S})$ still might render a reduced variance estimate of P_e. Recall that for Importance Sampling to be useful, we must have $\overline{W} \le P_e$. The conditions in (17) guaranteed this to be true, but they are certainly not necessary conditions. By estimating γ for systems whose marginal densities do not satisfy (17), we can approximately determine whether $\overline{W} \le P_e$. When γ^* is large, we can then assume we have a reduced variance estimate of P_e. We then can subsequently estimate the number of trials required for a particular confidence interval for P_e^*.

VI. Conclusions

In this paper, we have described Importance Sampling. This method will reduce the number of simulation trials for a fixed accuracy if the noise density is appropriately biased. We have shown that choosing $f_{\underline{R}^*}(\underline{R}) = f_{\underline{R}}(\underline{R} - \alpha\underline{S})$ for $0 \le \alpha \le 1$, will guarantee a reduced variance estimate of P_e for optimal detection systems under some conditions. The requirement is that the marginal density of the noise satisfy the conditions in (17). This choice of $f_{\underline{R}^*}(\underline{R})$ allows for very easy implementation on the computer since we do not need to generate a different type of random variable. The transformation from $\underline{R}$ to $\underline{R}^*$ is just $\underline{R}^* = \underline{R} + \alpha\underline{S}$. One complication of this biasing technique is that in most situations we are unable to find α_{opt}, but it appears from the cases in which we can obtain a solution that the choice $\alpha = 1/2$ is robust for small P_e.

To determine the performance gain of this biasing technique, we used the measure γ. This allowed us to solve for the reduction in trials over the standard Monte Carlo method as a function of P_e. Again, we can not always get analytical results for γ. We therefore showed one method of estimating this parameter. Using this estimate, we were able to show that γ^* is quite large for every system we investigated, and in certain systems, the number of trials can be made independent of the P_e. In general, it appears that using this biasing technique, we can substantially reduce the number of samples for a larger class of optimal detection systems than those restricted by conditions in (17).

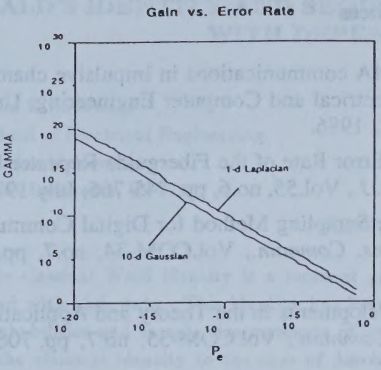


Figure (1). Gamma vs. Pe for 1-d Laplacian and 10-d Gaussian Systems.

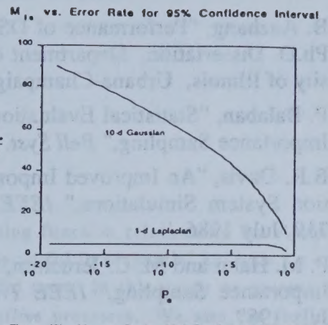


Figure (2). M_{Ie} vs. Pe for 95% Confidence Interval of [2 Pe/5, 8 Pe/5] for 10-d Gaussian and 1-d Laplacian Systems.

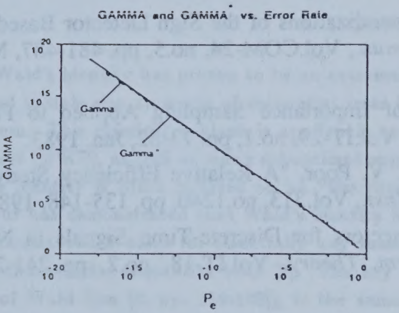


Figure (3). Gamma and Gamma* vs P_e for 1-d Laplacian Noise.

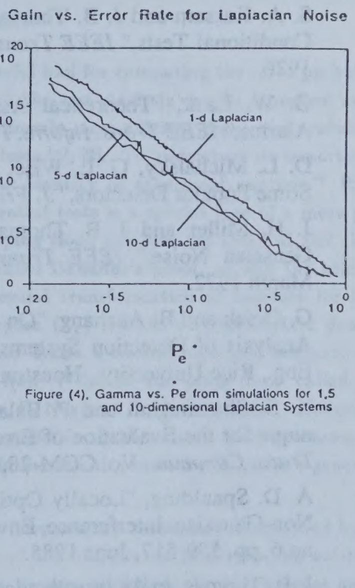


Figure (4). Gamma vs. Pe from simulations for 1,5 and 10-dimensional Laplacian Systems

References

[1] B. Aazhang, "Performance of DS/SSMA communications in impulsive channels," Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, IL, 1986.

[2] P. Balaban, "Statistical Evaluation of Error Rate of the Fiberguide Repeater Using Importance Sampling," *Bell Syst. Tech. J.*, Vol.55, no.6, pp. 745-766, July 1976.

[3] B.R. Davis, "An Improved Importance Sampling Method for Digital Communication System Simulations," *IEEE Trans. Commun.*, Vol.COM-34, no.7, pp. 715-719, July 1986.

[4] P. M. Hahn and M. C. Jeruchim, "Developments in the Theory and Application of Importance Sampling," *IEEE Tran. Commun.*, Vol.COM-35, no.7, pp. 706-714, July 1987.

[5] M. C. Jeruchim, "On the Application of Importance Sampling to the Simulation of Digital Satellite and Multihop Links," *IEEE Trans. Commun.*, Vol.COM-32, no.10, pp. 1088-1092, Oct. 1984.

[6] H. Kahn, "Use of Different Monte Carlo Sampling Techniques," in *Symposium on Monte Carlo Methods,* H. A. Meyer, Eds. Wiley: New York, 1956.

[7] S. A. Kassam and J. B. Thomas, "Generalizations of the Sign Detector Based on Conditional Tests," *IEEE Trans. Commun.*, Vol.COM-24, no.5, pp. 481-487, May 1976.

[8] G. W. Lank, "Theoretical Aspects of Importance Sampling Applied to False Alarms," *IEEE Trans. Inform. Theory*, Vol.IT-29, no.1, pp. 73-82, Jan. 1983.

[9] D. L. Michalsky, G. L. Wise, and H. V. Poor, "A Relative Efficiency Study of Some Popular Detectors," *J. Franklin Inst.*, Vol.313, no.1240, pp. 135-148, 1982.

[10] J. H. Miller and J. B. Thomas, "Detectors for Discrete-Time Signals in Non-Gaussian Noise," *IEEE Trans. Inform. Theory*, Vol.IT-18, no.2, pp. 241-250, March 1972.

[11] G. Orsak and B. Aazhang, "On the Theory of Importance Sampling Applied to the Analysis of Detection Systems," Tech. Rept. #8711, Dept. of Elec. and Comp. Eng., Rice University, Houston, TX, Sept. 1987.

[12] K. S. Shanmugam and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," *IEEE Trans. Commun.*, Vol.COM-28, no.11, pp. 1916-1924, NOV. 1980.

[13] A. D. Spaulding, "Locally Optimum and Suboptimum Detector Performance in a Non-Gaussian Interference Environment," *IEEE Trans. Commun.*, Vol.COM-33, no.6, pp. 509-517, June 1985.

[14] J. B. Thomas, in *An Introduction to Statistical Communication Theory,* John Wiley & Sons: New York, 1969.

[15] H. L. van Trees, *Detection, Estimation, and Modulation Theory, Part I.* New York: Wiley, 1968.

[16] B. Whittington, G. Orsak, and B. Aazhang, "Performance Analysis of Nonlinear Direct-Sequence Correlation Receivers," in *Proc. of 1987 Conf. on Inform. Sciences and Systems,* Baltimore, MD, The Johns Hopkins University, March 1987.

WALD'S IDENTITY AND SEQUENTIAL TEST PERFORMANCE WITH DEPENDENT DATA

John S. Sadowsky
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

The classical Wald identity is a moment generating function relationship for stopping times associated with i.i.d. data. This identity has been quite useful in sequential analysis for computing error probabilities and sample size moments of sequential tests. In this paper we present an extension of the classical identity to the case of *Markov-additive* processes. We also use the identity to prove an interesting *large deviations* limit theorem for sequential detectors: as the test thresholds tend to $\pm\infty$, the error probabilities decay exponentially and the exponential rate is identified. It is also demonstrated that our previous work on the robust modeling of data dependency is applicable. This provides a numerically efficient and accurate method of error probability computation which accounts for uncertain data dependency. A numerical example is provided.

I. INTRODUCTION

Wald's identity has proven to be an extremely powerful tool for computing the error probabilities and sample size moments of sequential tests for which the test statistic can be expressed as an i.i.d. sum. This classical analysis is applicable to the uniformly most efficient *sequential probability ratio test* (SPRT), as well as many suboptimal sequential tests [1], [2]. To underscore its importance, Wald's identity is often referred to as "the fundamental identity of sequential analysis." R. R. Bahadur has demonstrated that Wald's identity for sequential tests is a special case of a more general formula which holds for essentially any *random stopping time,* not just the interval exit times which characterize sequential tests [2]. The key idea behind Bahadur's proof, as well the original proof of Wald (see [1, pp. 159-160]), is the same exponential transformation of measure for i.i.d. sequences which is used in large deviations theory. Suppose $\{Z_k\}$ is i.i.d. with univariate density function $f(z)$ and $\mu(s) = \ln(E[\exp(sZ_k)]) < \infty$. Then we may define a transformed i.i.d. sequence distribution with univariate density $f^{(s)}(z) = \exp(sz-\mu(s))\,f(z)$. $f^{(s)}(z)$ is has often been called the exponentially *tilted* probability density function. Recently, P. Ney and E. Nummelin have developed a similar exponential measure transformation for *Markov-additive* (abbreviated MA) processes [3], [4], [5]. We have applied this MA process measure transformation to prove the Bahadur's generalization of the Wald identity for MA processes.

An MA process is simply an additive process with a "dependency structure" determined by an associated Markov chain. In this paper, the state space of the Markov chain may be an *arbitrary* measurable space. Thus, from the point of view of data process modeling, the MA process model admits a very broad class of ergodic sums. For example, stationary summands may be approximated by m'th order Markov chains. Another modeling option is the doubly stochastic process sometimes called a "hidden Markov model."

Section II presents preliminaries on MA processes, including the exponential measure transformation which is called the *conjugate MA process.* Section III presents the general form of Wald's identity. Section IV develops the application to the performance analysis of sequential tests. In addition to presenting formulas for computing error probabilities and expected sample sizes, we also

---

This work has been supported jointly by the National Science Foundation (contract DCI-8601071) and the Office of Naval Research (contract N00014-87-K-0155).

prove a large deviations theorem which indicates the exponential behavior of the error probabilities for tests with large thresholds. Section V shows how our previous work in [6] on robust modeling of uncertain data dependency applies to the analysis of sequential tests. To indicate the numerical feasibility of this method, Section V also gives a simple numerical example. Due to limited space, we have deleted all proofs.

II. PRELIMINARIES

Markov-Additive Processes

Let $\{X_k\}$ be a Markov chain taking values from a completely arbitrary measurable state space $(E,\mathcal{E})$. $\{Z_k\}$ shall be a process taking values in Euclidean space $\mathbf{R}^d$ which is associated with $\{X_k\}$ as follows: Let $\mathcal{B}^d$ denote the Borel field on $\mathbf{R}^d$. $(\Omega,\mathcal{F},\mathcal{P})$ is the underlying probability space, and $\mathcal{F}_n \triangleq \sigma(X_0,(X_1,Z_1),..,(X_n,Z_n))$. For any $A \in \mathcal{E}$ and $\Gamma \in \mathcal{B}^d$,

$$\mathcal{P}((X_{n+1},Z_{n+1}) \in A\times\Gamma \mid X_n = x,\mathcal{F}_n) = \mathcal{P}((X_{n+1},Z_{n+1}) \in A\times\Gamma \mid X_n = x) \triangleq P(x,A\times\Gamma). \quad (1)$$

This makes the pair $\{(X_k,Z_k)\}$ a Markov chain. Let $\pi_0(\cdot)$ (a probability measure on $\mathcal{E}$) denote the distribution of X_0. We shall always suppose that the probability distribution $\mathcal{P}(\cdot)$ on $(\Omega,\mathcal{F})$ may be specified by $P(\cdot,\cdot)$ and $\pi_0(\cdot)$. When $X_0 \equiv x$ (that is, when $\pi_0(\cdot) = \delta_x(\cdot)$), we shall write $\mathcal{P}_x(\cdot)$ instead of $\mathcal{P}(\cdot)$. Now define

$$S_n = \sum_{k=1}^{n} Z_k . \quad (2)$$

The pair $\{(X_n,S_n)\}$ is called a *Markov-additive* (abbreviated MA) process, and $P\colon E\times(\mathcal{E}\times\mathcal{B}^d) \rightarrow [0,1]$ in (1) is called the MA *transition probability.*

Consider some examples. If $Z_k = f(X_k,X_{k-1})$, then $\{(X_n,S_n)\}$ is an MA process. More generally, suppose that given the entire history $\mathcal{F}^X \triangleq \sigma(X_0,X_1,...)$, the process $\{Z_k\}$ is conditionally independent (but non-stationary) with $\mathcal{L}(Z_k|\mathcal{F}^X) = \mathcal{L}(Z_k|X_k,X_{k-1}) = F(\cdot\,|X_k,X_{k-1})$[1]. Then $\{(X_n,S_n)\}$ is an MA process. This is an example of a *doubly stochastic* MA process; the conditionally independent distributions of $\{Z_k\}$ are determined by sample paths of $\{X_k\}$.

Non-negative Kernels

Here we briefly review some key concepts from general non-negative kernel theory. For more detail, the interested reader will find Nummelin's book [7] to be an excellent reference. A *non-negative kernel* is a mapping $K\colon E\times\mathcal{E} \rightarrow [0,\infty]$, such that $K(\cdot,A)$ is a non-negative measurable function for each $A \in \mathcal{E}$, and $K(x,\cdot)$ is a non-negative measure for each $x \in \mathbf{E}$. As a operator we shall write $Kf(x) = \int f(y)\,K(x,dy)$. $K(\cdot,\cdot)$ is *irreducible* if there exists a measure $\phi(\cdot)$ such that for all $x \in \mathbf{E}$ and $A \in \mathcal{E}$ with $\phi(A) > 0$ we have $K^n(x,A) > 0$ for some integer $n < \infty$. It turns out that this general notion of irreducibility does not depend on the choice of $\phi(\cdot)$, as there always exists a "maximal irreducibility measure" (see Section 2.2 of [7]). An irreducible non-negative kernel $K(\cdot,\cdot)$ always has a *convergence parameter* $R = R(K)$. $R(K)$ is unique, well defined and $0 \leqq R(K) < \infty$. The precise definition is not required here (see [7, p. 27]); however, under appropriate conditions $R(K)^{-1}$ is the spectral radius. What is important for the present work is the following classification of kernels and the resulting "eigen-structure." In terms of $R = R(K)$, $K(\cdot,\cdot)$ may be classified as either *R-transient* or *R-recurrent* [7, §3.2]. If $K(\cdot,\cdot)$ is R-recurrent, there exists an *R-invariant function;* a positive measurable function $r(\cdot)$ such that $r(\cdot) = R\,Kr(\cdot)$ which is unique up to a scale factor (Theorem 5.1 in [7, p. 70]). In the R-recurrent case there also exists an *R-invariant measure;* a measure $\ell(\cdot)$ such that $\ell(A) = R\,\ell K(A) \triangleq \int K(x,A)\,\ell(dx)$ which is unique up to a scale factor (Theorem 5.2 in [7, p.73]). If $\int r(x)\,\ell(dx) < \infty$, then $K(\cdot,\cdot)$ is *R-positive recurrent* [7, p. 73]. In the R-transient case, there always exists an *R-subinvariant function;* a positive measurable function such that

---

[1] $\mathcal{L}(\cdot)$ denotes the *law,* or distribution, of the argument random quantity.

$r(\cdot) \geqq R\,Kr(\cdot)$ (Proposition 5.2 in [7, p. 69]). Subinvariants, however, may not be unique.

The Transform Kernel

We are interested in convergence parameters and invariant functions arising from a certain family of non-negative kernels called the *transform kernels.* These kernels are derived from an MA transition probability as follows: For each $s \in \mathbf{R}^d$, the transform kernel $\hat{P}(\cdot,\cdot;s)$: $\mathbf{E}\times\mathscr{E} \rightarrow [0,\infty]$ is defined by

$$\hat{P}(x,A;s) \;=\; \int_{\mathbf{R}^d} \exp(\, s\cdot z\,)\; P(x,A\times dz) \tag{3}$$

for all $x \in \mathbf{E}$ and $A \in \mathscr{E}$. (The $\cdot$ in (3) denotes the Euclidean inner product in $\mathbf{R}^d$.) It is convenient to define

$$\mu(s) \;\triangleq\; -\ln(\, R(\, \hat{P}(\,\cdot\,,\,\cdot\,;s)\,)\,) \tag{4}$$

(with $-\ln(0) = \infty$). The set $\mathscr{D} \triangleq \{\, s : \mu(s) < \infty \,\}$ is called the *essential domain* of $\mu(\cdot)$. We also define $\mathscr{D}_r \subset \mathscr{D}$ to be the set where the transform kernels $\hat{P}(\cdot,\cdot;s)$ are R-recurrent. For $s \in \mathscr{D}_r$, $r(\cdot;s)$ will denote the unique R-invariant function. We shall refer to $\{r(\cdot;s) : s \in \mathscr{D}_r\}$ as the family of *right eigenfunctions.* For $s \in \mathscr{D} - \mathscr{D}_r$, the transform kernel is R-transient, and in this case $r(\cdot;s)$ will denote an R-subinvariant function.

Hypothesis H: There exits a probability measure $\nu(\cdot)$ on $\mathscr{E}$, and a family of measures $\{h(x,\cdot)\colon x \in \mathbf{E}\}$ on $\mathscr{B}^{\,d}$, such that

$$h(x,\Gamma)\;\nu(A) \;\;\leqq\;\; P^n(x,A\times\Gamma) \tag{5}$$

for some $n < \infty$.

Lemma 1: Assume H. Then $\mu(\cdot)$: $\mathbf{R}^d \rightarrow (-\infty,\infty]$ is a convex lower semi-continuous function, $\mu(0) = 0$, and $\mathscr{D}$ is a convex subset of $\mathbf{R}^d$ containing $s = 0$.

Lemma 1 (which is proved in [5]) is the minimal characterization of $\mu(\cdot)$. Hypothesis H is quite mild; it amounts to little more than irreducibility of $\hat{P}(\cdot,\cdot;0)$. (See [4] for further discussion of H.) Under a slightly stronger hypothesis, much more is true. Typically both $\mu(s)$ and $r(\cdot;s)$ are analytic functions of s (again, see [4]). A strong recurrence hypothesis, called *uniform recurrence,* can also be used to prove these smoothness properties (analyticity, essential smoothness, etc.) [3]. For example, uniform recurrence always holds for irreducible MA processes on a finite state space.

The Conjugate MA Process

For any $s \in \mathscr{D}$, define

$$Q(x,dy\times dz;s) \;\triangleq\; \exp(\, s\cdot z - \mu(s)\,)\;\frac{r(y;s)}{r(x;s)}\;P(x,dy\times dz). \tag{7}$$

It turns out that $Q(\cdot,\cdot\times\cdot;s)$ is an MA transition probability with the same recurrence classification as the transform kernel $\hat{P}(\cdot,\cdot;s)$ [4], [5]. Thus (7) implicitly defines an *exponential measure transformation* of the probability distribution $\mathscr{P}(\cdot)$ to a new distribution $\mathscr{P}^{(s)}(\cdot)$.

Random Stopping Times

Let $\mathscr{F}_\infty = \sigma(X_0,(X_1,Z_1),(X_2,Z_2),\ldots)$. We say that an integer valued random variable $N \geqq 0$ is a *random stopping time* if, for each $n = 0,1,2,\ldots$, the conditional probability $\mathscr{P}(N = n \mid \mathscr{F}_\infty)$ is $\mathscr{F}_n$-measurable. That is, there exists Borel measurable functions α_n: $\mathbf{E}\times(\mathbf{E}\times\mathbf{R}^d)^n \rightarrow [0,1]$ such that

$$\mathscr{P}(\, N = n \mid \mathscr{F}_\infty\,) \;\;=\;\; \alpha_n(X_0,(X_1,Z_1),..,(X_n,Z_n)). \tag{8}$$

Furthermore, we suppose that these conditional probabilities do not depend on the MA transition probability $P(\cdot,\cdot)$ and initial distribution $\pi_0(\cdot)$. That is, the conditional probabilities (8) depend only on the sample paths of $\{(X_k,Z_k)\}$, and not on the MA distribution of $\{(X_k,Z_k)\}$.

A random stopping time implies that the distribution $\mathcal{P}(\cdot)$ on $(\Omega,\mathcal{F})$ is determined by the functions $\alpha_n(\cdot)$ as well as $P(\cdot,\cdot)$ and $\pi_0(\cdot)$. We may now formally define $\Omega = (E\times\{0,1\})\times(E\times R^d\times\{0,1\})^\infty$. This admits a $\{0,1\}$-valued process $\{U_k\}$ with conditional distributions determined by the sample paths of $\{(X_k,Z_k)\}$: $\mathcal{P}(U_n = 1 \mid \mathcal{F}_\infty) = \alpha_n(X_0,(X_1,Z_1),..,(X_n,Z_n))$. The random stopping time N can be identified as a (non-random) stopping time for the process $\{U_k\}$; $N = \inf\{n: U_n = 1\}$.

In applications, N is usually a stopping time for the sequence $\{Z_k\}$. For example, the stopping rule for a sequential test is the first exit time from an interval (a,b). It is obvious that this formulation also admits randomized stopping rules.

III. WALD'S IDENTITY FOR MA PROCESSES

Theorem 1: Assume H and $X_0 = x$. Let N be any random stopping time. Then for any $s \in \mathcal{D}$,

$$\mathcal{P}_x^{(s)}(N < \infty) = \frac{E_x[\exp(s\cdot S_N - N\mu(s))\, r(X_N;s) \mid N < \infty]}{r(x;s)}\, \mathcal{P}_x(N < \infty). \tag{9}$$

Corollary 1: Assume H, and $\mathcal{L}(X_0) = \pi_0$. Suppose that for some $s \in \mathcal{D}$ we have $E[r(X_0;s)] < \infty$, and hence we may define the probability measure $\pi_0(dx;s) = r(x;s)\pi_0(dx)/E[r(X_0;s)]$. Let $\mathcal{P}^{(s)}(\cdot)$ be the probability distribution determined by the conjugate probability (7) and $\mathcal{L}^{(s)}(X_0) = \pi_0(\cdot;s)$. Then

$$\mathcal{P}^{(s)}(N < \infty) = \frac{E[\exp(s\cdot S_N - N\mu(s))\, r(X_N;s) \mid N < \infty]}{E[r(X_0;s)]}\, \mathcal{P}(N < \infty). \tag{10}$$

In the next section, we shall show that for the application to sequential tests, $\mathcal{P}(N < \infty) = \mathcal{P}^{(s)}(N < \infty) = 1$. In this case, (10) simplifies to

$$E[\exp(s\cdot S_N - N\mu(s))\, r(X_N;s)] = E[r(X_0;s)]. \tag{10'}$$

We note, however, that the general form of (10) is useful in other applications. For example, in the analysis of the Fano and stack algorithms, (which are used to decode convolutional codes) Wald's identity is applied to estimate a probability $\mathcal{P}(N < \infty) < 1$.

In the case that $\{Z_k\}$ is i.i.d., it turns out that $r(\cdot;s) \equiv 1$ for all $s \in R^d$. In this case, the above formulas reduce to the classical identity of [2].

Some General Expressions for Stopping Time Expectation

Assume now that $d = 1$; that is, the additive component of the MA process is one dimensional. We can immediately apply the above results to get some general formulas for the expectation E[N]. Suppose that by some means it can be demonstrated that the following conditions hold. There exists an open interval $I \subset \mathbf{R}$ such that $0 \in I$, and for all $s \in I$, (i) $\mathcal{P}^{(s)}(N < \infty) \equiv 1$, (ii) $E[r(X_0;s)] < \infty$, and (iii) $\mu(s)$ and $r(x,s)$ are differentiable with respect to s, and differentiation may be interchanged with expectation. We note that (i) and (iii) are practically equivalent to $I \subset \mathcal{D}$, and $\mathcal{P}(N < \infty) = 1$. Also, these assumptions are almost automatic under the uniform recurrence condition. We shall denote differentiation with respect to s with a prime ('). Assuming (i) - (iii), we may differentiate (10') with respect to s and evaluate at $s = 0$. Assuming that $\mu'(0) \neq 0$, and using $\mu(0) = 0$ and $r(\cdot;0) \equiv 1$, a little manipulation gives

$$E[N] = \frac{E[S_N] + E[r'(X_N;0) - r'(X_0;0)]}{\mu'(0)}. \tag{11}$$

It is important to note that this formula holds only when the eigenfunction family is normalized so that $r(\cdot;0) \equiv 0$. If $\mu'(0) = 0$, then a formula for E[N] may be obtained by differentiating twice.

Similar differentiation arguments will also also yield general formulas for higher order moments of N.

IV. APPLICATION: PERFORMANCE ANALYSIS FOR SEQUENTIAL TESTS

Sequential Tests

The primary application of Wald's identity is in the analysis of sequential tests. The most notable example is the ***sequential probability ratio test*** (SPRT) which is uniformly most efficient for i.i.d. data [1]. The analysis here is directly applicable to SPRT with simple hypotheses of Markov data. However, we are not restricted to the SPRT. We emphasize this because the present analysis may be applied to find the worst-case error probabilities of a given sequential test with *compound* statistical hypotheses. Throughout this section we consider only one dimensional test statistics ($d = 1$) although our general formulation clearly admits multidimensional test statistics.

For our purposes, a sequential test is simply the observation of the first exit of S_n from an interval (a,b) where $-\infty < a < 0 < b < \infty$. The stopping time N, which is the time of first exit, is sample size required for a decision. If S_n exits (a,b) above ($S_N \geqq b$) then the decision is in favor of the statistical hypothesis H_1. Likewise, if S_n exits below ($S_N \leqq a$) then the decision is in favor of H_0. It turns out that the "drift" of S_n is determined by $\mu'(0)$, which we assume exists. This follows from large deviations results for MA processes; S_n/n converges in probability to $\mu'(0)$, and hence, $E[\,S_n\,] \sim n\,\mu'(0)$. Any reasonable sequential test will tend to have S_n exit (a,b) below for H_0, and above for H_1. Thus, in general, we shall characterize the statistical hypotheses as follows: we assume that the interior of $\mathscr{D}$ is nonempty and contains 0, $\mu(\cdot)$ is differentiable at 0, and

$$\begin{aligned} &H_1:\ \mu'(0) > 0 \qquad (\text{ positive drift }) \\ &H_0:\ \mu'(0) < 0 \qquad (\text{ negative drift })\,. \end{aligned} \tag{12}$$

The following lemma allows us to use the identity (10').

Lemma 2: Assume H. If $\{X_k\}$ is recurrent Markov chain, and $\mu'(0) \neq 0$, then $\mathscr{P}(\,N < \infty\,) = 1$. More generally, if $s \in \mathscr{D}_r$ and $\mu'(s) \neq 0$, then $\mathscr{P}^{(s)}(\,N < \infty\,) = 1$.

Error Probabilities

The error probabilities for a sequential test are

$$P_{e0} \triangleq \mathscr{P}\,(\text{error}|H_0) = \mathscr{P}\,(\,S_N \geqq b\,) \tag{13a}$$

and

$$P_{e1} \triangleq \mathscr{P}\,(\text{error}|H_1) = \mathscr{P}\,(\,S_N \leqq a\,). \tag{13b}$$

Since $\mathscr{P}(\cdot)$ is determined by $P(\cdot,\cdot)$ and $\pi_0(\cdot)$, in order to consider compound hypotheses we may think of these error probabilities as functionals of $P(\cdot,\cdot)$ and $\pi_0(\cdot)$; $P_{ei} = P_{ei}(P,\pi_0)$.

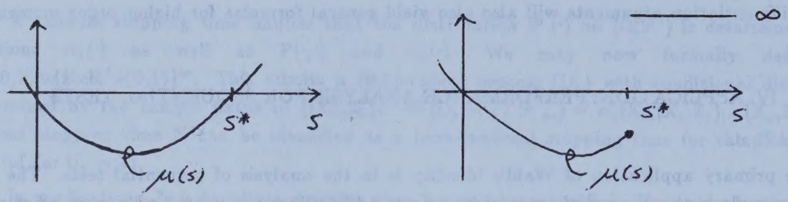
By Lemma 1, $\mu(\cdot)$ is a lower semi-continuous convex function, and clearly, $0 \in \mathscr{D}$. For an H_0 distribution, define

$$s^* \triangleq \sup\{\,s : \mu(s) \leqq 0\,\}, \tag{14a}$$

and for an H_1 distribution,

$$s^* \triangleq \inf\{\,s : \mu(s) \leqq 0\,\}. \tag{14b}$$

The lower semi-continuity of $\mu(\cdot)$ implies $\mu(s^*) \leqq 0$ because $\{s: \mu(s) \leqq 0\}$ is closed. Subject to condition (12), we can have $s^* = 0$ only if 0 is a boundary point of $\mathscr{D}$. In particular, if $\mathscr{D}$ has a non-empty interior containing $s = 0$, then $s^* \neq 0$. By condition (12), we have $s^* > 0$ for H_0 and $s^* < 0$ for H_1. Under H_0, the determination of s^* is illustrated in Figure 1 for the two possible cases; $\mu(s^*) = 0$ and $\mu(s^*) < 0$.



a) $\mu(s^*) = 0$ b) $\mu(s^*) < 0$

Figure 1: Determination of s^* under H_0.

Definition: We shall say an MA transition probability $P(\cdot,\cdot)$ is *regular* if, in addition to hypothesis H, we have (i) $s^* \neq 0$, (ii) $\mu(s^*) = 0$ (as in Figure 1a), and (iii) $\mathcal{P}(N < \infty) = \mathcal{P}^{(s^*)}(N < \infty) = 1$.

Define

$$r_a(s) \triangleq \frac{E[\exp(s(S_N - a))\, r(X_N;s) \mid S_N \leqq a]}{E[r(X_0;s)]} \tag{15a}$$

and

$$r_b(s) \triangleq \frac{E[\exp(s(S_N - b))\, r(X_N;s) \mid S_N \geqq b]}{E[r(X_0;s)]}. \tag{15b}$$

Lemma 3: Suppose that $P(\cdot,\cdot)$ is a regular MA transition probability and $E[r(X_0;s^*)] < \infty$. Then

$$P_{e0} = \frac{1 - r_a(s^*)\exp(s^*a)}{r_b(s^*)\exp(s^*b) - r_a(s^*)\exp(s^*a)}, \tag{16a}$$

and

$$P_{e1} = \frac{1 - r_b(s^*)\exp(s^*b)}{r_a(s^*)\exp(s^*a) - r_b(s^*)\exp(s^*b)}. \tag{16b}$$

The next result is a Chernoff bound for sequential tests. It holds with essentially no recurrence hypothesis (although the conclusion is trivial when $s^* = 0$).

Lemma 4: Assume H, and $E[r(X_0;s^*)] < \infty$ (the MA transition probability need not be regular). Then under H_0, $0 \leqq s \leqq s^*$, and under H_1 for any $s^* \leqq s \leqq 0$, we have

$$P_{e0} \leqq \frac{\exp(-sb)}{r_b(s)} \quad \text{and} \quad P_{e1} \leqq \frac{\exp(-sa)}{r_a(s)}. \tag{17a,b}$$

Now consider a comparison of (16) and the bound (17) with $s = s^*$. In (16a), since $s^* > 0$ for H_0, if $b - a$ is moderately large, then the terms $r_a(s^*)\exp(s^*a) \ll 1$ will be negligible in comparison to $r_b(s^*)\exp(s^*b) \gg 1$. By neglecting $r_a(s^*a)$, (16a) reduces to the right side of (17a). The point is that *the upper bounds* (17) *are tight when* $P(\cdot,\cdot)$ *is regular.* This is particularly true when the thresholds are large in magnitude, and hence the error probabilities are small. Now, suppose that it can be demonstrated that $r_a(s^*)$ and $r_b(s^*)$ vary slowly with a and b. Then the error probabilities are essentially determined by thresholds a and b and the parameter s^*, which depends on $P(\cdot,\cdot)$ but not on a and b. That is, $P_{e0} \sim \exp(-s^*b)$ and $P_{e1} \sim \exp(-s^*a)$. We formalize this statement in

Theorem 2, which is a large deviations theorem for sequential tests.

As in [4] and [5], we obtain a general upper bound in Theorem 2 by modifying the error probabilities with the argument "$X_N \in A$" where A is a "small set"; for example, we replace P_{e0} by $\mathscr{P}(S_n \geqq b, X_N \in A)$. This is still a strong theorem as there always exists a sequence of small sets $A_i \uparrow E$ [5]. If the entire state space E is small, then the additional argument may be deleted altogether. Uniform recurrence is a sufficient (but not necessary) condition for E to be small.

Definition: For a fixed $s \in \mathscr{D}$, a set $A \in \mathscr{E}$ is a *small set* if for some constant $k > 0$, $k \leqq r(x;s)$ for all $x \in A$.

Theorem 2: Assume H and $E[\, r(X_0;s^*)\,] < \infty$. Consider just H_0 ($\mu'(0) < 0$). Then for any s^* small set A,

$$\overline{\lim_{b \uparrow \infty}} \frac{1}{b} \ln(\mathscr{P}(S_N \geqq b, X_N \in A)) \leqq -s^*. \tag{18a}$$

Furthermore,

$$\underline{\lim_{b \uparrow \infty}} \frac{1}{b} \ln(P_{e0}) \geqq -s^*. \tag{18b}$$

Similar bounds hold under H_1; replace b by $-a$, $S_N \geqq b$ by $S_N \leqq a$, and $-s^*$ by s^*.

Small Excess Approximations

In many cases, when S_n crosses one of the thresholds at time N, the difference between S_N and the threshold is negligible. That is, either $S_N \approx a$ or $S_N \approx b$. These are called the *Wald,* or *small excess* approximations [1]. Applying the small excess approximations to (16a) and (16b), we have

$$r_a(s) \approx \hat{r}_a(s) \triangleq \frac{E[\, r(X_N;s) \mid S_N \leqq a\,]}{E[\, r(X_0;s)\,]} \tag{19a}$$

and

$$r_b(s) \approx \hat{r}_b(s) \triangleq \frac{E[\, r(X_N;s) \mid S_N \geqq b\,]}{E[\, r(X_0;s)\,]}. \tag{19b}$$

These approximations may be applied to (17) or (18) to obtain useful error probability approximations. In a more rigorous fashion, if Z_k is bounded, we may obtain upper and lower error probability bounds. These bounds will be tight if $\| Z_k \|_\infty$ is small in comparison to the threshold, that is, if $\| Z_k \|_\infty$ is a small excess.

Another reasonable approximation for weak dependency is $r(\cdot;s) \approx 1$ (recall for i.i.d. data $r(\cdot;s) \equiv 1$). In combination with the small excess approximation, we then have $r_a(s) \approx r_b(s) \approx 1$.

V. LEAST FAVORABLE DEPENDENCY STRUCTURE

In Section IV, we have shown that error probabilities depend on the MA transition probability primarily through the parameter s^*. This principle was formalized by Theorem 2. The practical conclusion is: *MA transition probabilities which minimize* $|s^*|$ *are "least favorable"* in the sense that they asymptotically maximize error probabilities. This principle is now applied to a problem of robust error probability computations with uncertain dependency structure. Loosely speaking, assume that the data is approximately i.i.d., yet some degree of data dependency is known to be present. The precise structure of the dependency is not known. To model this situation, we construct a nominally i.i.d. *dependency contamination class* of MA process distributions which preserves the (assumed known) univariate distribution.

Consider *separable* MA transition probabilities; that is, MA probabilities of the form

$$P(x,A\times\Gamma) \;=\; F(\Gamma|x)\,P(x,A) \tag{20}$$

where P(x,A) is just the transition probability of the Markov chain $\{X_k\}$ and $\{F(\,\cdot\,|x) : x \in E\}$ is a family of conditional probability distributions on **R**. In general, this is a doubly stochastic model; $\mathscr{L}\,(Z_k|X_{k-1} = x) = F(\,\cdot\,|x)$. Of course, if $Z_{k+1} = g(X_k)$, then $F(\,\cdot\,|x) = \delta_{g(x)}(\cdot)$. The sequential "dependency structure" of the observed data process $\{Z_k\}$ is modeled by $P(\cdot,\cdot)$. We suppose that $P(\cdot,\cdot)$ is positive recurrent with stationary distribution $\pi(\cdot)$. X_0 will also have distribution $\pi(\cdot)$, so it follows that $\{X_k\}$ and $\{Z_k\}$ are jointly stationary. The transform kernels for separable MA probabilities of the form (20) reduce to

$$\hat{P}(x,A;s) \;=\; \exp(\,\hat{g}(x;s)\,)\,P(x,A) \tag{21}$$

where the function $\hat{g}(\cdot;s)$ is given by

$$\hat{g}(x;s) \;=\; \ln(\,E[\,\exp(sZ_{k+1})\,|\,X_k = x\,]\,) \;=\; \ln(\,\textstyle\int \exp(sz)\,F(dz|x)\,). \tag{22}$$

If $Z_{k+1} = g(X_k)$, then $\hat{g}(x;s) = sg(x)$.

The admissible degree of data dependency will be controlled by a one shift ***maximal autocorrelation*** bound:

$$\sup_{g(\cdot)} \frac{|\,\mathrm{cov}[\,g(X_k)\,,\,g(X_{k+1})\,]\,|}{\mathrm{var}[\,g(X_k)\,]} \;\leqq\; \rho \tag{23}$$

where $0 < \rho < 1$ and the maximization is over all real valued functions $g(\cdot)$ such that $0 < \mathrm{var}[\,g(X_k)\,] < \infty$. Since (23) depends only on $\{X_k\}$, it is a constraint on $P(\cdot,\cdot)$.

Finally, define

$$\tilde{P}_\rho(x,A) \;=\; \rho\,\delta_x(A) \;+\; (1-\rho)\,\pi(A). \tag{24}$$

This is the transition probability of the *Markov jump process* with stationary distribution $\pi(\cdot)$ and dependency parameter ρ. We call this a (discrete-time) jump process because $\{X_k\}$ will maintain constant values on runs of random length. From (24), $\mathscr{P}\,(\,X_k = X_{k-1}\,) = \rho$ (when $\pi(\cdot)$ is non-atomic), and hence, the run length distribution is geometric with parameter ρ. It is also easy to verify that $\tilde{P}_\rho(\cdot,\cdot)$ satisfies (23). In the notation, a tilde (˜) and the subscript ρ will denote quantities associated with the jump process.

Definition: Let $\mathscr{C}_{\pi,\rho}$ denote the class of all Markov probabilities $P\colon E\times\mathscr{E} \to [0,1]$ such that (i) $P(\cdot,\cdot)$ is positive recurrent with stationary distribution $\pi(\cdot)$, (ii) $P(\cdot,\cdot)$ satisfies the maximal correlation bound (23) with parameter ρ, and (iii) the associated transform kernel $\hat{P}(\cdot,\cdot;\tilde{s}^*_\rho)$ is $R(\tilde{s}^*_\rho)$-recurrent.

In condition (iii), note that the recurrence condition for $\mathscr{C}_{\pi,\rho}$ is determined by the jump process. We note, for example, that all uniformly recurrent Markov probabilities satisfying (i) and (ii) are in $\mathscr{C}_{\pi,\rho}$. Hence, this is a rich nonparametric class of process distributions. We call $\mathscr{C}_{\pi,\rho}$ a *dependency contamination class.* It models the uncertain sequential dependency in the observed data process $\{Z_k\}$ through the underlying unknown Markov chain distributions of $\{X_k\}$. The stationary distribution $\pi(\cdot)$ and the family $\{F(\,\cdot\,|x) : x \in E\}$ are assumed known. So, $\mathscr{C}_{\pi,\rho}$ represents uncertainty only with regard to sequential dependency. We note, however, that uncertainty in $\pi(\cdot)$ and $\{F(\,\cdot\,|x) : x \in E\}$ can be easily handled using the usual methods (ϵ-contamination models, etc.). We have proven the following theorem in [6].

Theorem 3: Suppose that $\hat{g}(\cdot;\tilde{s}^*_\rho)$ is a bounded function. Then $\tilde{P}_\rho \in \mathscr{C}_{\pi,\rho}$, and for all $P \in \mathscr{C}_{\pi,\rho}$ we have $|s^*| \geqq |\tilde{s}^*_\rho|$. Furthermore, we may obtain $\tilde{\mu}_\rho(s)$ as the solution of

$$E\left[\frac{1-\rho}{\exp(\ \tilde{\mu}_\rho(s) - \hat{g}(X;s)\ ) \ - \ \rho} \right] \ = \ 1 \tag{25}$$

where X is an E-valued random variable with distribution $\pi(\cdot)$, and

$$\tilde{r}_\rho(x;s) \ = \ \frac{1-\rho}{\exp(\ \tilde{\mu}_\rho(s) - \hat{g}(x;s)\ ) \ - \ \rho} \tag{26}$$

An Example: The Sequential Sign Test

Suppose that real valued data $\{X_k\}$ is observed and we wish to test

$$H_0\text{: median } X_k < 0 \quad \text{vs.} \quad H_1\text{: median } X_k > 0.$$

The sequential sign test, with thresholds a and b, is specified by $Z_k = \text{sgn}(X_k)$. Assume that $\mathscr{P}\,(\,X_k = 0\,) = 0$. Then Z_k is a ± 1 valued random variable with $\mathscr{P}\,(\,Z_k = +1\,) = \mathscr{P}\,(\,X_k \geqq 0\,) \triangleq p$. First note that $\hat{g}(x;s) = s\ \text{sgn}(x) = \pm s$. It is a very simple matter to solve (25) to compute $\tilde{\mu}_\rho(s)$ and $\tilde{s}^*_\rho$ using standard numerical methods (see [6]). From (26), $\tilde{r}_\rho(x;s)$ depends on x only through sgn(x), hence, $E[r(X_0;s)]$ is easily computed in terms of the parameter p. Also, given $S_N \geqq b$ we must have $\text{sgn}(X_N) = +1$. Likewise, $\text{sgn}(X_N) = -1$, given $S_N \leqq a$. These facts easily determine $\tilde{r}_{\rho,a}(\tilde{s}^*_\rho)$ and $\tilde{r}_{\rho,b}(\tilde{s}^*_\rho)$. Finally, we note that if the thresholds a and b are integers, then there is no excess; either $S_N = a$ or $S_N = b$. Thus, Wald's small excess approximations are exact, and we may easily compute the exact error probabilities $\tilde{P}_{ei}$, i=0,1. The expected sample size E[N] is also easily computed using equation (11).

For a performance comparison, we shall also consider $X_k = \theta + N_k$ where $\{N_k\}$ is Gaussian autoregressive noise with one shift autocorrelation ρ. (For a Gaussian process, we can prove that the autocorrelation agrees with the maximal autocorrelation (23).) We have estimated the sign detector error probabilities with the autoregressive noise using Monte Carlo simulation. Table 1 presents this comparison for thresholds a = 10 and b = −10. The simulations were run until the empirical standard deviation was approximately 1/10th of the error probability estimate.

The comparison presented in Table 1 indicates that the jump process error probability performance is only slightly pessimistic in comparison to the autoregressive process simulations. This indicates that the jump process can be a very good model for dependency. The expected sample size for the jump process, on the other hand, is consistently less than occurred in the autoregressive simulations. However, to compare expected sample size, we must adjust the threshold to obtain equal error probabilities. For example, with $\theta = 0.6$ and $\rho = 0.5$, compare the jump process performance with thresholds ±11 against the autoregressive simulations with thresholds ±10. The jump process error probability is $1.7{\times}10^{-2}$, which is close to the simulation result, $1.5{\times}10^{-2}$. The jump process expected sample size is 22.4, which compares quite well with the simulation result, 24.4.

VI. CONCLUSION

In this paper we have presented a general form of Wald's identity which accounts for sequential data dependency. We have also developed in some detail the application to sequential test performance. The example given in Section V was quite simple; however, this method of analysis does provide a numerically tractable means for computing error probabilities of more complex detector structures.

The general form of the identity presented in Section III is also useful for other applications. In a future publication, we shall apply this identity to the analysis of the *stack algorithm* (also known as the *Z-J algorithm,* after Zigangirov and Jelinek). This is a sequential algorithm for searching random trees which was originally developed for decoding convolutional codes. We have applied the Wald identity to extend the notion of a *computational cutoff* condition. When operated "above" the cutoff condition, the expected computational requirement of the stack algorithm becomes infinite, while "below" the condition there is a finite upper bound. Previous developments of this classical result have required i.i.d. assumptions and special random trees generated by a "Fano metric." Our analysis will eliminate both of these conditions.

Table 1: Comparison of Jump Process Performance Computations With Gaussian Autoregressive Simulations

			Error Probability		Expected Sample Size	
θ	p	ρ	Simulation	Jump Process	Simulation	Jump Process
0.4	.3446	0.2	5.8×10^{-3}	8.4×10^{-3}	32.6	29.5
		0.5	5.9×10^{-2}	7.2×10^{-2}	32.6	25.8
0.5	.3085	0.2	2.4×10^{-3}	2.6×10^{-3}	26.5	24.4
		0.5	3.1×10^{-2}	4.0×10^{-2}	27.9	22.6
0.6	.2743	0.2	6.4×10^{-4}	8.5×10^{-4}	22.5	20.9
		0.5	1.5×10^{-2}	2.2×10^{-2}	24.4	20.5
0.7	.2420	0.2	1.7×10^{-4}	2.8×10^{-4}	19.7	18.4
		0.5	7.1×10^{-3}	1.3×10^{-2}	21.6	18.0

REFERENCES

[1] A. Wald, *Sequential Analysis.* New York: Dover, 1973. New York: Wiley, 1947.

[2] R. R. Bahadur, "A note on the fundamental identity of sequential analysis," *Ann. Math. Statis.*, vol. 29, pp. 534-543, 1958.

[3] I. Iscoe, P. Ney, and E. Nummelin, "Large deviations of uniformly recurrent Markov-additive processes," *Adv. Appl. Math.*, vol. 6, pp. 373-412, 1985.

[4] P. Ney and E. Nummelin, "Markov-additive processes I: eigenvalue properties and limit theorems," *Ann. Prob.*, vol. 15, pp. 593-609, 1987.

[5] P. Ney and E. Nummelin, "Markov-additive processes II: large deviations," *Ann. Prob.*, vol. 15, pp. 561-592, 1987.

[6] J. S. Sadowsky, "An asymptotically least-favorable Chernoff bound for a large class of dependent data processes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 52-61, 1987.

[7] E. Nummelin, *General Irreducible Markov Chains and Non-Negative Operators.* Cambridge: Cambridge University Press, 1984.

DISTRIBUTED DETECTION WITH CORRELATED SENSOR NOISE

R. VISWANATHAN, V. AALO AND S. C. A. THOMOPOULOS
Department of Electrical Engineering
Southern Illinois University
Carbondale, Illinois 62901

ABSTRACT

The effect of correlated noise process on the performance of a distributed detection system is studied. We have assumed that the local sensors have the same operating point, and that the distribution of the sensor observations is symmetric, thus implying that the joint distribution of the sensor decisions and, therefore, the fusion rule are symmetric functions of the sensor decisions. As an example, we consider the detection of a constant signal in additive Gaussian noise with three sensors. Positive correlation coefficients decrease system performance while negative correlations improve the performance. Though the OR-rule of fusing the sensor decisions is relatively insensitive to the changes in the correlation coefficients, it performs poorly compared to AND and MAJORITY Logic rules.

This research is sponsored by the SDIO/IST and managed by the Office of Naval Research under Contract N00014-86-0515.

INTRODUCTION

In distributed detection systems, local sensors make decisions based on their observed data and transmit their decisions to a fusion center where the global decision is made. Thus there is a considerable amount of savings in communication bandwidth in not sending the observations to the fusion center. However, since all the information is not available at the fusion center there is a loss of performance in distributed detection systems. In many applications the use of multiple sensors is also more attractive due to considerations of cost, reliability, and survivability.

Distributed detection has received a lot of attention in the literature in recent years [1-7]. The study of distributed detection was pioneered by Tenney and Sandell [1] who extended classical Bayesian theory to distributed testing, without formulating a data fusion rule. Ekchian and Tenney considered distributed detection for different network topologies [2]. Sandaji [3] studied the problem of hypothesis testing in a distributed environment while Chair and Varshney [4] obtained the optimum fusion rule for the case of distributed sensors in which the operating points are known and fixed. Srinivasan [5] extended the results in [4] to show that the optimum thresholds for the local sensors are not only coupled with one another, but they also depend on the fusion rule. By sending some quality bits along with the decisions of the local sensors, instead of sending crisp decisions only, it has been shown that the system performance improves considerably [6,7]. Optimal partitioning of the observations at the local sensors has also been considered. In this case a network of n sensors receiving independent observations transmit these partitioned information to the fusion center. It has been shown that for a two hypotheses test carried out at the fusion center, the partitioning based on the likelihood of the observations maximizes the probability of detection for a given probability of false alarm at the fusion center [8].

In all the approaches above, the local sensors have been assumed to operate independently of each other. It has thus been usually assumed that the observation vector (and hence the decisions of the local sensors) are conditionally, given the hypothesis, independent of each other. The independence assumption is reasonable in problems of detection of known signals in independent noise. Other cases may arise, however, where the assumption of statistical independence is no longer valid, for example in the detection of unknown signals in noise that may arise in sonar problems [1]. Here we study the effects of correlated noise at the local sensors on the performance of a distributed detection system. We extend the results obtained in [5] to the case where the noise at the local sensors are correlated. Each local sensor performs a likelihood ratio test, operating under identical thresholds. If we do not impose the identicality of the operating points, the solution to the problem seems interstable (see N-P completeness for the case of discrete observations, [9]).

DISTRIBUTED DETECTION WITH CORRELATED SENSOR NOISE

Consider the two hypothesis testing with distributed sensors shown in Fig. 1. The two hypotheses to be tested are:

H_0: Target absent
H_1: Target present

The observed data at each local sensor, Z_i ($i = 1, 2, \ldots, N$) is given by

$$\underline{Z}_i = \begin{cases} \underline{n}_i & \text{under } H_0 \\ \underline{n}_i + \underline{S} & \text{under } H_1 \end{cases}$$

where the noise and signal vectors $\underline{n}_i$ and $\underline{S}$ are such that the observation vector cannot be taken to be independent from sensor to sensor. The decisions of the local sensors are denoted by $u_1, u_2, \ldots, u_N$ where

$$u_i = \begin{cases} 0 & \text{when no target is detected} \\ 1 & \text{when a target is detected.} \end{cases}$$

Also the joint density of the sensor decisions under each hypothesis cannot be taken to be independent. Using the decisions of the local sensors as its own data, the fusion center performs the following likelihood ratio test in order to make the global decision:

$$\Lambda(\underline{u}) = \frac{P(u_1, u_2, \ldots, u_N/H_1)}{P(u_1, u_2, \ldots, u_N/H_0)} \overset{H_1}{\underset{H_0}{\gtrless}} \lambda \qquad (1)$$

where λ is the threshold at the fusion center. In order to show that the likelihood rule implies a symmetric fusion rule, we rewrite the likelihood ratio as

$$\Lambda(\underline{u}) = \frac{\int_{u_1}\int_{u_2}\cdots\int_{u_N} p_{Z_1 \ldots Z_N}(z_1 \ldots z_N/H_1)\,dz_N \ldots dz_1}{\int_{u_1}\int_{u_2}\cdots\int_{u_N} p_{Z_1 \ldots Z_N}(z_1 \ldots z_N/H_0)\,dz_N \ldots dz_1} \qquad (2)$$

where $u_1\ u_2 \ldots u_N$ on the integral signs denote the appropriate region of integration, depending on whether $u_i = 0$ or 1. If the tests at the sensors are identical (i.e., they all use the same threshold) and the distribution of the sensor observations are symmetric, then the value of the integrals in the numerator and denumerator of (2) do not change if any of the variables are permutated. Thus the fusion rule must be a symmetric function of the sensor decisions.

NUMERICAL EXAMPLE

To illustrate the effects of correlated noise on the performance of the distributed detection system in Fig. 1, we consider three local sensors that receive equicorrelated zero mean Gaussian noise. For a specified probability of false alarm at the fusion center, we compute the required common threshold at the local sensors and then obtain the probability of detection (P_{D_0}) as a function of the correlation coefficient, which can be positive or negative. Each of the three symmetric fusion rules will be considered separately.

OR Strategy

With this type of combining rule, the fusion center decides in favor of H_0 (target absent) only when all the three local sensors decide that target is absent. Since the local sensors operate at the same threshold, t, the false alarm probability at the fusion center (α_0) can be written as

$$\begin{aligned} \alpha_0 &= 1 - P(u_1 = 0,\ u_2 = 0,\ u_3 = 0/H_0) \\ &= 1 - \int_{-\infty}^{t}\int_{-\infty}^{t}\int_{-\infty}^{t} p_Z(\underline{Z}/H_0)\, d\underline{Z} \end{aligned} \tag{3}$$

where $p_Z(\underline{Z}/H_0)$ is the density of the sensor observations under the hypothesis H_0 and is a function of the correlation coefficient, ρ.

(i) $\underline{0 \leq \rho < 1}$:

Using the results in [11,12], equation (3) becomes

$$\alpha_0 = 1 - \int_{-\infty}^{\infty} F^3\left(\frac{t - \sqrt{\rho}y}{\sqrt{1 - \rho}}\right) f(y)\, dy \tag{4}$$

where f(y) and F(x) are the standard normal density and CDF respectively. Given α_0, we solve eq. (4) for the threshold, t and obtain the probability of detection at the fusion center as

$$P_{D_0} = 1 - \int_{-\infty}^{\infty} F^3\left(\frac{t - s - \sqrt{\rho}y}{\sqrt{1 - \rho}}\right) f(y)\, dy \tag{5}$$

where s is the signal level at each local sensor.

(ii) $\underline{-0.5 < \rho \leq 0}$:

For negative correlation coefficients, equation (3) becomes [11,13]

$$\alpha_0 = 1 - P(u_1 = 0,\ u_2 = 0,\ u_3 = 0/H_0) = 1 - F_3(t/\rho) \tag{6}$$

where $F_3(\ )$ is defined in [13], and

$$P_{D_0} = 1 - P(u_1 = 0,\ u_2 = 0,\ u_3 = 0/H_1) = 1 - F_3(t\text{-}s/\rho) \tag{7}$$

AND Strategy

In this combining rule, the fusion center decides in favor of hypothesis H_1 only when all the three sensors decide H_1. Again using the results in [11] we can compute the common threshold (t) from which we can obtain P_{D_0} for given values of α_0 and ρ.

(i) $0 \leq \rho < 1$:

In this case the false alarm probability is given by

$$\alpha_0 = P(u_1 = 1, u_2 = 1, u_3 = 1/H_0) \tag{8}$$

$$= \int_{-\infty}^{\infty} [1 - F(\frac{t - \sqrt{\rho}y}{\sqrt{1 - \rho}})]^3 f(y)\, dy$$

Similarly

$$P_{D_0} = P(u_1 = 1, u_2 = 1, u_3 = 1/H_1) \tag{9}$$

$$= \int_{-\infty}^{\infty} [1 - F(\frac{t - s - \sqrt{\rho}y}{\sqrt{1 - \rho}})]^3 f(y)\, dy$$

(ii) $- 0.5 < \rho \leq 0$:

For negative correlation coefficients, we have

$$\alpha_0 = 1 - 3F_1(t/\rho) + 3F_2(t/\rho) - F_3(t/\rho) \tag{10}$$

$$P_{D_0} = 1 - 3F_1(t\text{-}s/\rho) + 3F_2(t\text{-}s/\rho) - F_3(t\text{-}s/\rho) \tag{11}$$

MAJORITY Logic

In this combining rule, the global decision is in favor of H_1 when two or more of the three local sensors decide that the target is present.

(i) $0 \leq \rho < 1$:

The probability of false alarm in this case is given by

$$\alpha_0 = 3 \int_{-\infty}^{\infty} F(\frac{t - \sqrt{\rho}y}{\sqrt{1 - \rho}})[1 - F(\frac{t - \sqrt{\rho}y}{\sqrt{1 - \rho}})]^2 f(y)\, dy \tag{12}$$

$$+ \int_{-\infty}^{\infty} [1 - F(\frac{t - \sqrt{\rho}y}{\sqrt{1 - \rho}})]^3 f(y)\, dy$$

from which we obtain the threshold and compute the probability of detection.

$$P_{D_0} = 3\int_{-\infty}^{\infty} F\left(\frac{t - s - \sqrt{\rho}y}{\sqrt{1-\rho}}\right)\left[1 - F\left(\frac{t - s - \sqrt{\rho}y}{\sqrt{1-\rho}}\right)\right]^2 f(y)\,dy + \int_{-\infty}^{\infty}\left[1 - F\left(\frac{t - s - \sqrt{\rho}y}{\sqrt{1-\rho}}\right)\right]^3 f(y)\,dy \tag{13}$$

(ii) $-0.5 < \rho \leq 0$:

For negative values of ρ, we have the following equations

$$\alpha_0 = 1 + 2F_3(t/\rho) - 3F_2(t/\rho) \tag{14}$$

$$P_{D_0} = 1 + 2F_3(t\text{-}s/\rho) - 3F_2(t\text{-}s/\rho) \tag{15}$$

Figures 2-5 show plots of the probability of detection against the correlation coefficient for the three fusion rules. Each graph is plotted for a specific value of α_0 and different signal-to-noise ratios. It can be seen from these graphs that the OR-combining rule, though performs poorly compared to the AND and MAJORITY rules, appears to be insensitive to changes in the correlation coefficient. In general negative correlation between the sensor noises drastically improves the performance of the system.

CONCLUSION

We have studied the effect of correlated noise on the performance of a distributed detection system. We assumed that the joint distribution of the sensor observations is a symmetric function of the variables and that all the local sensors operate at the same point. It is shown that the joint distribution of the sensor decisions and the combining rule must by symmetric functions of the variables. As an example, the detection of a constant signal in additive Gaussian noise with three sensors is considered.

The performance of the system with the OR strategy is relatively insensitive to the correlation between the sensor noise. Also negative correlation improves the system performance while positive correlation decreases system performance for all the three fusion rules.

The analysis of distributed detection with arbitrarily correlated sensor observations seems extremely complicated. A similar problem, assuming discrete observations is known to be N-P complete [9].

REFERENCES

[1] R. R. Tenney and N. R. Sandell, Jr., "Detection with Distributed Sensors," Trans. Aerospace and Electronic Systems, Vol. AES-17, pp. 501-510, July 1981.

[2] L. K. Ekchian and R. R. Tenney, "Detection Networks," Proceedings of the 21st IEEE Conference on Decision and Control, Orlando, Fla., pp. 686-691, Dec. 1982.

[3] F. A. Sadjadi, "Hypothesis Testing in a Distributed Environment," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-22, No. 2, pp. 134-137, March 1986.

[4] Z. Chair and P. K. Varshney, "Optimal Data Fusion Multiple Sensor Detection Systems," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-22, pp. 98-101, Jan. 1986.

[5] R. Srinivasan, "Distributed Radar Detection Theory," IEE Proceedings, Vol. 133, Pt. F, pp. 55-60, Feb. 1986.

[6] S. C. A. Thomopoulos, R. Viswanathan and D. P. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," Proceedings in Multiple Sensor Systems," Proceedings of 24th Annual Allerton Conference on Communication, Control, and Computing, pp. 984-993, Oct. 1986.

[7] J. J. Chao and C. C. Lee, "A Distributed Detection Scheme Based on Soft Local Decisions," Proceedings of the 24th Annual Allerton Conference on Communications, Control, and Computing, pp. 974-983, Oct. 1986.

[8] R. Viswanathan, A. Ansari and S. C. A. Thomopoulos, "Optimal Partitioning of Observations in Distributed Detection," Submitted to IEEE Transaction on Information Theory.

[9] J. N. Tsitsiklis and M. Athans, "On the Complexity of Decentralized Decision Making and Detection Problems," IEEE Transactions on Automatic Control, Vol. AC-30, No. 5, pp. 440-446, May 1985.

[10] A. R. Reibman and L. W. Nolte, "Optimal Detection and Performance of Distributed Sensor Systems," IEEE Trans. Aerospace and Electronic System, Vol. AES-23, No. 1, pp. 24-30, Jan. 1987.

[11] V. Aalo, R. Viswanathan and S. C. A. Thomopoulos, "A Study of Distributed Detection with Correlated Sensor Noise," Submitted to Globecom, 1987.

[12] S. S. Gupta, "Probability Integrals of Multivariety Normal and Multivariate t," Annals of Math, Statistics, Vol. 34, pp. 792-828, Sept.-Dec. 1963.

[13] G. P. Steck and D. B. Owen, "A Note on the Equicorrelated Multivariate Normal Distribution," Biometrika, Vol. 49, pp. 269-271, 1962.

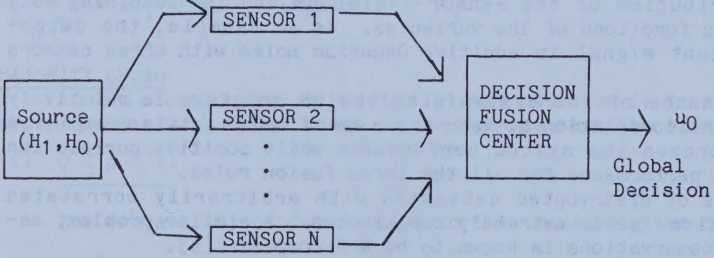


Figure 1: Illustration of Decision Fusion

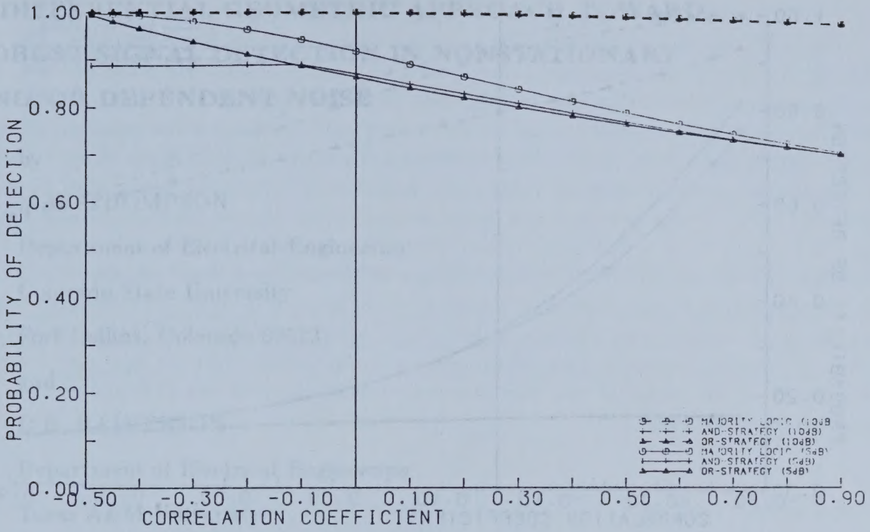


FIG 2 : P_{D_0} VS ρ α_0 = 1.0E-1

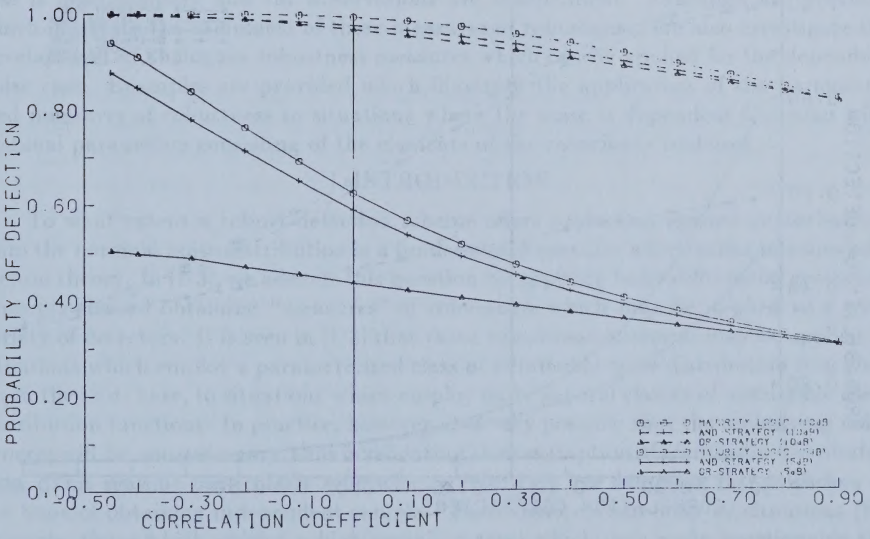


FIG 3: P_{D_0} VS ρ α_0 = 1.0E-2

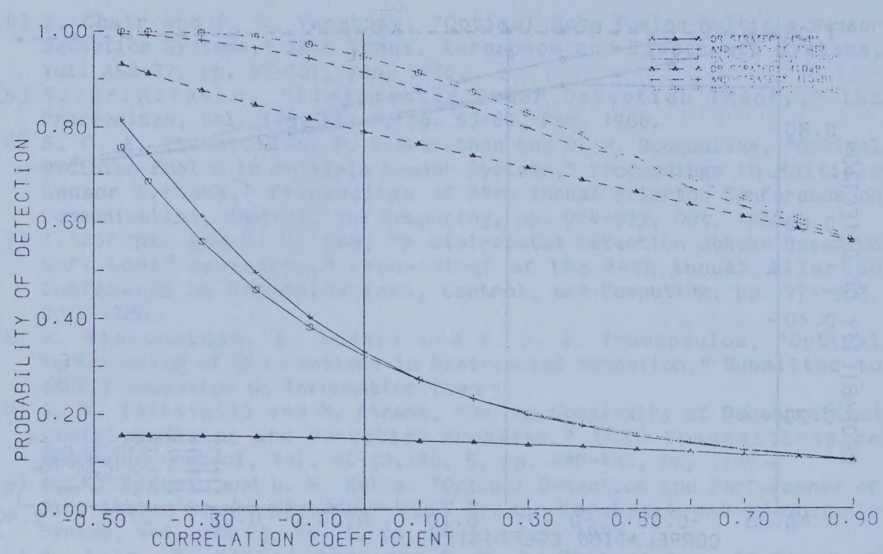


FIG.4 : P_{D_0} VS ρ $\alpha_0 = 1.0E-3$;

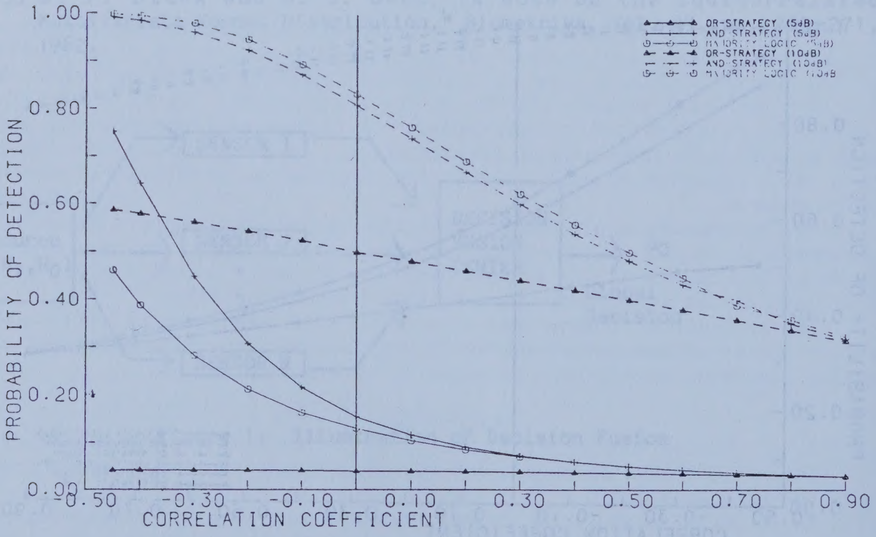


FIG. 5 : P_{D_0} VS ρ $\alpha_0 = 1.0E-4$

A DIFFERENTIAL GEOMETRIC APPROACH TOWARD ROBUST SIGNAL DETECTION IN NONSTATIONARY AND/OR DEPENDENT NOISE

by

M.W. THOMPSON

Department of Electrical Engineering

Colorado State University

Fort Collins, Colorado 80523

and

D.R. HALVERSON

Department of Electrical Engineering

Texas A&M University

College Station, Texas 77843

ABSTRACT

We present the development of robustness measures (both local and nonlocal) which are based on geometric concepts and are applicable to situations where the noise process is nonstationary and the observations are independent. Examples are provided which illustrate the usefulness of these measures of robustness. We also investigate the development of analogous robustness measures which can be applied for the dependent noise case. Examples are provided which illustrate the application of the parameterized measures of robustness to situations where the noise is dependent Gaussian with nominal parameters consisting of the elements of the covariance matrices.

I. INTRODUCTION

To what extent a robust detection scheme offers protection against perturbations from the nominal noise distribution is a fundamental question which arises in robust detection theory. In [1-3], we address this question by applying basic differential geometric concepts toward obtaining "measures" of robustness which may be applied to a wide variety of detectors. It is seen in [1-3] that these robustness measures may be applied to situations which employ a parameterized class of admissible noise distribution functions or, in the i.i.d. case, to situations which employ quite general classes of admissible noise distribution functions. In practice, however, it is very possible that the underlying noise process will be nonstationary, thus invalidating the assumption of identically distributed data. This may be particularly true when a relatively low sampling rate is chosen in the hope of obtaining independent samples. Furthermore, there may be situations (for example, those which employ a high sampling rate) which may make questionable the independence assumption. Often classical saddlepoint robustness techniques such as those found in [4-6] are difficult to apply to situations where the noise is nonstationary and /or dependent. In this paper we investigate the development of robustness measures which are similar to those in [1-3], yet may be applied to the nonstationary and /or dependent noise case. Such robustness measures could be useful in developing robust detection schemes beyond those which are tractable using saddlepoint methods.

II. DEVELOPMENT

In [1-3], we view the robustness problem from a geometric perspective. Let D_n denote the class of n-dimensional distribution functions. Form this point of view, the performance of the detector may be expressed by considering the performance functional $h : D_n \to R$. For example, h could be associated with the false alarm probability α or the detection probability β. In the case where h is associated with β, we wish to choose a detector such that h is reasonably high and does not vary much near the nominal element of D_n. Thus, viewing h as a height function over D_n, we could say that a robust detector would yield a "surface" above which the nominal element is both relatively high and not strongly sloped. By employing a differential geometric approach in [1], we obtain a measure of robustness for classes of admissible underlying distributions which are parameterized by m parameters. In this parameterized situation, we associate D_n with R^m and our measure is based upon associating the slope of the unit normal with the cosine of the angle of the unit normal to vertical. We recall from [1] that for parameters $a_1, \cdots, a_m$

$$\phi_{2,m} = \cos\gamma_m = \frac{1}{\sqrt{1+\sum_{i=1}^{m}(\frac{\partial h}{\partial a_i})^2}},$$

where γ_m denotes the angle between the unit normal and vertical. Note that to achieve a greater degree of robustness, we should choose a detector such that $||(\frac{\partial h}{\partial a_1}, \cdots, \frac{\partial h}{\partial_m})||_2$ is as small as possible. This observation motivates another natural measure of robustness which is obtained by replacing the l_2 norm with l_1. This leads to the measure of robustness

$$\phi_{1,m} = \frac{1}{1+\sum_{i=1}^{m}|\frac{\partial h}{\partial a_i}|}.$$

In [1], we considered the situation where the noise is i.i.d. and extended the $\phi_{2,m}$ measure of robustness to apply to a more general class of admissible noise distribution functions. Taking an approach similar to [1], we can likewise extend the $\phi_{1,m}$ measure of robustness and yet relax the assumption on the noise to simply independence (not necessarily identically distributed). The employment of the ℓ_1 norm was shown in [7] to have some subtle but important advantages over the ℓ_2 norm and will therefore be used here. We note that the above definition of $\phi_{1,m}$ admits nonstationary data whose n-dimensional distribution function is parameterized by m parameters. Let $X_1, X_2, \cdots, X_n$ be independent random variables which represent the sampled data. Furthermore, let $F_1, F_2, \cdots, F_n$ be the nominal distribution functions which correspond to $X_1, X_2, \cdots, X_n$, respectively. We recall that a detector of interest can be characterized by a decision region represented by a Borel set B_n in R^n, which corresponds to the set of observations for which we announce H_1. We then have

$h = \int_{B_n} dF_1(x_1) \cdots dF_n(x_n)$, where (in the common case where randomization is not necessary) $h = \alpha$ if the $F_i(\cdot)$ are governed by H_0, and $h = \beta$ if the $F_i(\cdot)$ are governed by H_1. As in [1], for $i = 1, 2 \cdots, n$, we define

$$\begin{aligned}\partial_i^+ \beta_n = \{(x_1, x_2, \cdots, x_n) :\ & \text{there exists } \varepsilon > 0 \text{ such that} \\ & (x_1, \cdots, x_{i-1}, y, x_{i+1}, \cdots, x_n) \in \bar{B}_n \text{ for } y \in (x_i - \varepsilon, x_i) \text{ and} \\ & (x_1, \cdots, x_{i-1}, z, x_{i+1}, \cdots, x_n) \in B_n \text{ for } z \in (x_i, x_i + \varepsilon)\},\end{aligned}$$

$$\begin{aligned}\partial_i^- B_n = \{&(x_1, x_2, \cdots, x_n) : \text{there exists } \varepsilon > 0 \text{ such that} \\ &(x_1, \cdots, x_{i-1,}, y, x_{i+1}, \cdots, x_n) \in B_n \text{ for } y \in (x_i - \varepsilon, x_i) \text{ and} \\ &(x_1, \cdots, x_{i-1}, z, x_{i+1}, \cdots, x_n) \in \bar{B}_n \text{ for } z \in (x_i, x_i + \varepsilon)\} \quad .\end{aligned}$$

Choosing a rectilinear partition P of R^n, we note that in order to extend the parameterized robustness measure to a generalized robustness measure it suffices, as in [1-3], to limit consideration to distribution functions $F_i(\cdot)$ given by step functions, i.e. functions of form

$F_i(\cdot) = \sum_{j=1}^{m_i} a_{ij} I_{A_{ij}}(\cdot)$, where the m_i are natural numbers, the a_{ij} are reals between zero and unity, and the A_{ij} are the bounded subsets of the reals which generate P. Letting m denote the total number of bounded partitioning subsets of R^n, we note that the a_{ij} parameterize $F_1(\cdot), \cdots, F_n(\cdot)$, and we thus define the general robustness measure

$$\phi = \lim_{|P| \to 0} \phi_{1,m} ,$$

where (as is the custom for extended integrals) the existence of the limit corresponds to the same value being obtained regardless of how the union of the bounded partitioning sets approaches R^n. We then have (defining $a_{i, m_i+1} = 1$ for $i = 1, \cdots, n$)

$$h = \sum_{j_1=1}^{m_1} \cdots \sum_{j_n=1}^{m_n} \prod_{i=1}^{n} (a_{i,j_i+1} - a_{i,j_i})[I_{B_n}(x_{1,j_1}, \cdots, x_{n,j_n})]$$

where $x_{i,j_i} = \sup A_{i,j_i}$ for $i = 1, \cdots, n$ and $j_i = 1, \cdots, m_i$. We thus obtain for $k = 1, \cdots, n$ and $\ell = 2, \cdots, m_k$

$$\frac{\partial h}{\partial a_{k,\ell}} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_{k-1}=1}^{m_{i-1}} \sum_{j_{k+1}=1}^{m_{i+1}} \cdots \sum_{j_n=1}^{m_n} \prod_{\substack{i=1 \\ i \neq k}}^{n} (a_{i,j_i+1} - a_{i,j_i})$$

$$\begin{aligned}\bullet[&I_{B_n}(x_{1,j_1}, \cdots x_{k-1,j_{k-1}}, x_{k,\ell-1}, x_{k+1,j_{k+1}}, \cdots, x_{n,j_n}) \\ &-I_{B_n}(x_{1,j_1}, \cdots, x_{k-1,j_{k-1}}, x_{k,\ell}, x_{k+1,j_{k+1}}, \cdots, x_{n,j_n})].\end{aligned}$$

Suppose that if $x \notin \partial_1^+ B_n \cap \cdots \cap \partial_n^+ B_n$ then $x \epsilon$ int $(B_n) \cup \text{int}(\bar{B}_n)$. For the common situation where $\partial_i^- B_n = \phi$ for $i = 1, \cdots, n$, we then have

$$\phi = \frac{1}{1 + \sum_{k=1}^n |\int_{\partial_k^+ B_n} dF_1(x_1) \cdots dF_{k-1}(x_{k-1}) dF_{k+1}(x_{k+1}) \cdots dF(x_n)|} ,$$

where $F_k(\cdot)$ is the nominal distribution of X_k.

As an example of the application of the above expression for the robustness measure ϕ, consider first the employment of the linear detector, where for a set of observations $\{y_i; i = 1, 2, \cdots, n\}$ and positive real numbers b_i the test statistic $\sum_{i=1}^n b_i y_i$ is compared to a threshold T. This leads to a decision region B_n with the associated sets (for $i = 1, 2, \cdots, n$)

$$\partial_i^+ B_n = \{(y_1, y_2, \cdots, y_n) : \sum_{i=1}^n b_i y_i = T\}$$

$$\partial_i^- B_n = \phi \quad \cdot$$

We therefore obtain

$$\phi_{\text{linear}} = \frac{1}{1+n} .$$

Note that as the number of samples n approaches infinity then ϕ_{linear} approaches zero; thus, the linear detector becomes completely "unrobust" as the number of samples approaches infinity.

On the other hand, consider the Huber-type detector which employs a censored form of the linear detector, wherein a test statistic of form $\sum_{i=1}^{n} g_i(y_i)$ is compared to a threshold T, with

$$g_i(x) \begin{cases} b_i x & \text{if } |b_i x| \le k_i \\ k_i & \text{if } \; b_i x > k_i \;, \\ -k_i & \text{if } \; b_i x < -k_i \end{cases}$$

where b_i and k_i are positive real numbers. For $n = 2$ we can succinctly specify the relevant sets

$$\begin{aligned} \partial_1^+ B_2 &= \{(y_1, y_2) : y_1 = (T-k_2)/b_1 \;,\; y_2 > k_2/b_2; \\ &\qquad \text{or } b_1 y_1 + b_2 y_2 = T, \;\; (T-k_2)/b_1 \le y_1 < k_1/b_1\} \\ \partial_2^+ B_2 &= \{(y_1, y_2) : y_2 = (T-k_1)/b_2, \; y_1 > k_1/b_1; \\ &\qquad \text{or } b_1 y_1 + b_2 y_2 = T, \;\; (T-k_2)/b_1 < y_1, \le k_1/b_1\} \\ \partial_1^- B_2 &= \partial_2^- B_2 = \phi \quad \cdot \end{aligned}$$

We then obtain for $n = 2$

$$\phi_{\text{ Huber },n=2} = \frac{1}{1 + [1 - F_1((T-k_2)/b_1)] + [1 - F_2((T-k_1)/b_2)]}$$

A similar but considerably more lengthy analysis then shows for arbitrary n,

$$\phi_{\text{Huber}} = \frac{1}{1 + \sum\limits_{i=1}^{n} \prod\limits_{\substack{j=1 \\ j \ne i}}^{n} [1 - F_j((T - \sum\limits_{\substack{p=1 \\ p \ne j}}^{n} k_p)/b_j)]} \;\cdot$$

We therefore observe that $\phi_{\text{Huber}} \ge \phi_{\text{linear}}$.

Consider now the situation where we fix a desired false alarm rate $\alpha > 0$. If we are interested in the robustness of the actual false alarm rate, then the $F_i(\cdot)$ are considered to be under H_0 and examination of the integral expression for α shows

$$\prod_{j=1}^{n} [1 - F_j((T - \sum_{\substack{p=1 \\ p \ne j}}^{n} k_p)/b_j)] \ge \alpha \quad \cdot$$

We therefore obtain

$$\phi_{\text{Huber}} \le \frac{1}{1 + n\alpha} ,$$

and thus as for the linear detector, $\phi_{\text{Huber}} \to 0$ as $n \to \infty$. We therefore conclude that while the Huber-type detector is no less robust than the linear one, this advantage atrophies completely as the number of samples approaches infinity.

Similar conclusions hold under H_1. If we are interested in the robustness of a detector's detection probability β_n (subscripted to reflect dependence on the number of samples), we obtain the same expression as before for the Huber-type detector, with β_n replacing α. Since β_n is typically lower bounded by α, we again have in this situation $\phi_{\text{Huber}} \to 0$ under H_1. Moreover, for the case where the nominal distributions $F_i(\cdot)$ are commonly equal to a distribution function $F(\cdot)$, and thus the k_i and b_i are constant, we can say more. In this case we have under H_1,

$$\phi_{\text{Huber}} \leq \frac{1}{1 + n\beta_n^{(n-1)/n}}.$$

Now for the very common situation where $\beta_n \to 1$ as $n \to \infty$ we have $\frac{\phi_{\text{Huber}}}{\phi_{\text{linear}}} \to 1$ as $n \to \infty$, and thus not only does the Huber-type detector, like the linear detector, become completely unrobust as the number of samples approaches infinity, but the rate at which this happens is the same as for linear detector. Note that this latter situation still admits the presence of nonstationary perturbations of the nominal distribution; we only assume the nominal element is the same for $i = 1, 2, \cdots, n$.

The aforementioned expressions for the measure of robustness ϕ are derived from a local approach toward robustness; the conclusions are applicable as the unknown distributions vary over small neighborhoods about the corresponding nominals. However, these results can be extended to much larger neighborhoods by taking a "worst case" approach. Suppose we consider for $k = 1, 2, \cdots, n$ the family $\bar{F}_k$ of distribution functions which are lower bounded by a distribution function $G_k(\cdot)$. The measure of "worst case" robustness over $\{\bar{F}_k\}_{k=1}^n$ may thus be defined by

$$\phi_{\text{inf}} = \inf\{\phi(F_1, F_2, \cdots, F_n) : F_k \varepsilon \bar{F}_k \text{ for } k = 1, 2, \cdots, n\},$$

where $\phi(F_1, F_2, \cdots, F_n)$ is the previously defined ϕ. Taking note of the aforementioned expressions for ϕ for the linear detector and the Huber-type detector, we observe that for those detectors under the same regularity conditions,

$$\phi_{\text{inf}} = \frac{1}{1 + \sum_{k=1}^{n} \left| \int_{\partial_k^+ B_n} dG_1(x_1) \cdots dG_{k-1}(x_{k-1}) dG_{k+1}(x_{k+1}) \cdots dG_n(x_n) \right|}.$$

Moreover, the same expression is obtained for many other detectors. For example, if a detector possesses a decision region B_n with $\partial_i^- B_n = \phi$ for $i = 1, 2, \cdots, n$, and if for $i = 1, 2, \cdots, n$ there exist real numbers $c_{i,1}, c_{i,2}, \cdots, c_{i,n-1}$ and continuous functions

$$H_i : (c_{i,1}, \infty) \times (c_{i,2}, \infty) \times \cdots \times (c_{i,n-1}, \infty) \to R$$

such that

$$\partial_i^+ B_n = \{(x_1, x_2, \cdots, x_n) : x_i = H_i(x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)\},$$

then the above expression for ϕ_{inf} is applicable to this detector, for essentially the same reason as for the linear and Huber-type detectors. We therefore have available for a large family of detectors an easily evaluated expression for the nonlocal measure of robustness ϕ_{inf} with application to independent but not identically distributed data.

III. DEPENDENT PROCESSES

The techniques of [1] directly apply for the case of parameterized dependent processes. If the joint distribution corresponding to n-samples is parameterized by m parameters, $a_1, a_2, \cdots, a_m$, the same expression as given earlier provides a measure of robustness, i.e.,

$$\phi_{1,m} = \frac{1}{1 + \sum_{i=1}^{m} \left|\frac{\partial h}{\partial a_i}\right|}, \quad \text{where}$$

h may be chosen either as false alarm probability or detection probability. We now illustrate the application of this technique via an example.

Consider the discrete time detection of a constant, positive signal s in dependent zero-mean Gaussian noise with the nominal parameters being the elements of the covariance matrix Λ. Thus, we consider a hypothesis test of the following form:

$$\begin{aligned} H_0 &: X_i = N_i \quad , i = 1, 2, \cdots, n \\ &\qquad\qquad\qquad\qquad , \\ H_1 &: X_i = N_i + s \; , \; i = 1, 2, \cdots, n \end{aligned}$$

where $\{N_i\}_{i=1}^{n}$ are jointly Gaussian , zero mean random variables with a nominal covariance matrix Λ. For the three sample case, we shall compare the performance and robustness of the canonical memoryless linear detector to the performance and robustness of the Huber-type detector, which is obtained by censoring the linear detector at the vertical heights $-k$ and k. Often censoring of this nature is employed in the hope of imparting robustness.

First consider the linear detector. The threshold T_L is chosen such that for a false alarm rate α,

$$\alpha = P\left(\sum_{i=1}^{3} N_i \geq T_L\right) \quad .$$

Note that the joint density of $\{N_i\}_{i=1}^{3}$ is given by

$$f_{\underline{N}}(\underline{x}) = \frac{1}{(2\pi)^{3/2}|\Lambda|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\underline{x}^T \Lambda^{-1} \underline{x}\right\} \quad .$$

It is then straightforward to show that

$$\alpha = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{T_L - x_1 - x_2}^{\infty} f_{\underline{N}}(\underline{x}) dx_1 dx_2 dx_3 .$$

Furthermore, the detection probability is given by

$$\beta_L = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{T_L - x_1 - x_2}^{\infty} f_{\underline{N}}(\underline{x} - \underline{s}) dx_1 dx_2 dx_3 ,$$

where $\underline{s}^T = (s, s, s)$ and $\underline{x}^T = (x_1, x_2, x_3)$.

Lengthy but straightforward calculations then yield expressions for $\frac{\partial \beta_L}{\partial_{\sigma_{ij}}}$, $i, j = 1, 2, 3$, where σ_{ij} is the i, j *th* element of Λ. These expressions are too cumbersome to display here, but bear a passing resemblance to those found in [1] for the parameterized

Gaussian example. We then have that the robustness measure for the linear detector is given by

$$\phi_{1,3}^{L} = \frac{1}{1+\sum\limits_{i=1}^{3}|\frac{\partial\beta_i}{\partial\sigma_{ii}}|+\sum\limits_{\substack{i=1\\ i\neq j}}^{3}\sum\limits_{j=1}^{3}|\frac{\partial\beta_L}{\partial\sigma_{ij}}|} .$$

Evaluating this expression using the nominal parameter values $\sigma_{11} = \sigma_{22} = \sigma_{33} = 1$, $\sigma_{12} = \sigma_{21} = \sigma_{23} = \sigma_{32} = 0.5$, and $\sigma_{13} = \sigma_{31} = 0.25$, we have for a false alarm rate $\alpha = 0.05$ and signal strength $s = 0.5$ that $\beta_L = 0.157$ and $\phi_{1,3}^{L} = 0.83\cdot$

A similar approach may be used when employing the Huber-type detector. For this detector, α is given by

$$\alpha = \int\limits_{T_H-2k}^{\infty}\int\limits_{T_H-2k}^{\infty}\int\limits_{T_H-2k}^{\infty} f_{\underline{N}}(\underline{x})d\underline{x} ,$$

$$-\int\limits_{T_H 2_k}^{4k-T_H}\int\limits_{T_H-2k}^{2k-x_1}\int\limits_{T_H-2k}^{T_H-x_1-x_2} f_{\underline{N}}(\underline{x})d\underline{x}$$

where T_H is the threshold chosen to achieve α. It also follows that

$$\beta_H = \int\limits_{T_H-2k}^{\infty}\int\limits_{T_H-2k}^{\infty}\int\limits_{T_H-2k}^{\infty} f_{\underline{N}}(\underline{x}-\underline{s})d\underline{x}$$

$$-\int\limits_{T_H-2k}^{4k-T_H}\int\limits_{T_H-2k}^{2k-x_1}\int\limits_{T_H-2k}^{T_H-x_1-x_2} f_{\underline{N}}(\underline{x}-\underline{s})d\underline{x} \cdot$$

Again, lengthy calculations yield expressions for $\frac{\partial\beta_H}{\partial\sigma_{ij}}$, $i,j = 1,2,3$.

In a manner analogous to the linear detector we then may compute the robustness measure $\phi_{1,3}^{H}$ for the Huber-type detector. Evaluating at the aforementioned nominal parameter values, we have for $\alpha = 0.05$, $s = 0.5$, and $k = 1$ that $\beta_H = 0.141$ and $\phi_{1,3}^{H} = 0.78$.

We naturally would be interested in extending the above analysis for the dependent parameterized case to a more general setting, as has been demonstrated in [1] for the independent case. However, difficulties arise when attempting to do this. We can illustrate the type of obstacles encountered by focusing on the two sample case. We then have $h = \int_{B_2} dF_{X_1,X_2}(x_1,x_2)$, which motivates limiting consideration to

$$h = \sum_{i=1}^{m}\sum_{j=1}^{m}(a_{i+1,j+1} - a_{i,j+1} - a_{1+1,j} + a_{i,j})I_{B_2}(x_i,y_j),$$

where the (x_i,y_j) are vertices of partitioning rectangles of R^2 and the $a_{i,j}$ are nonnegative real numbers. We then obtain

$$\frac{\partial h}{\partial a_{k,\ell}} = I_{B_2}(x_{k-1},y_{\ell-1}) - I_{B_2}(x_k,y_{\ell-1}) - I_{B_2}(x_{k-1},y_\ell) + I_{B_2}(x_k,x_\ell) ,$$

and then find that $\sum_{k=1}^{m} \sum_{\ell=1}^{m} |\frac{\partial h}{\partial a_{k,\ell}}|$ diverges as the norm of the partition approaches zero for many Borel sets B_2. For example, this occurs for a Borel set which is obtained by perturbing into a chain of stair steps the boundary of the set corresponding to the linear detector. The resultant Borel set, no matter how small the width of the steps, leads to divergence of $\sum_{k=1}^{m} \sum_{\ell=1}^{m} |\frac{\partial h}{\partial a_{k,\ell}}|$. Interpreting the consequences of this divergence somewhat freely, we might be led to conclude that there are so many possible ways of varying a multivariate distribution function about a nominal that our measure of robustness points toward complete unrobustness, even for small numbers of samples. A modification of these techniques would therefore appear desirable to draw any further conclusions.

IV. CONCLUSIONS

In this paper we employ geometric concepts to develop robustness measures (both local and nonlocal) which, unlike the measures of [1-3], may be applied to situations where the noise process is nonstationary and the observations are independent. Examples are given which illustrate the application of these robustness measures. In addition, we consider the situation where there exists dependency between observations. Examples are provided which illustrate the application of the parameterized robustness measures where we employ a dependent Gaussian noise distribution with the nominal parameter values being the entries of the covariance matrix. We also investigate extending the $\phi_{1,m}$ parameterized robustness measure to obtain a more general robustness measure for the dependent noise case. It is seen that attempting to extend the measure for the dependent case in a manner which is analogous to the independent case results in convergence problems for some situations.

ACKNOWLEDEMENT

This research was supported by the Air Force Office of Scientific Research under Grant AFOSR-87-0087.

REFERENCES

1. M.W. Thompson and D.R. Halverson, "A novel differential geometric approach toward robust signal detection," *Proceedings of the 1986 Conf. on Information Sciences and Systems*, Princeton, New Jersey, March 19-21, 1986, pp. 11-16.
2. M.W. Thompson and D.R. Halverson, "Aspects of robust signal detection from a differential geometric viewpoint," *Proceedings of the Twenty-Fourth Annual Allerton Conf. on Communication, Control and Computing*, Monticello, Illinois, Oct. 1-3, 1986, pp. 236-245.
3. M.W. Thompson and D.R. Halverson, "A geometric measure of robustness applied to a nonstandard class of detectors," *Proceedings of the 1987 Conference on Information Sciences and Systems*, Baltimore, Maryland, March 25-27, 1987, pp. 348-353.
4. P.J. Huber, "Robust estimation of a local parameter," *Ann. Math Stat.*, vol. 35, pp. 73-101, March 1964.
5. P.J. Huber, "A robust version of the probability ratio test," *Ann. Math. Stat.*, vol. 36, pp. 1753-58, December 1965.
6. P.J. Huber, *Robust Statistics*, Chap. 10. New York: Wiley, 1981.
7. M.W. Thompson, *Applications of Geometric Concepts Toward Robust Signal Detection* (Ph.D. dissertation), Chap. 3. Texas A&M University, August 1987.

Presented at the Twenty-Fifth Annual Allerton Conference on Communication, Control, and Computing, September 30 - October 2, 1987; to be published in the Proceedings of the Conference.

SIGNAL DETECTION IN ARCTIC UNDER-ICE NOISE

P.A. NIELSEN and JOHN THOMAS
Princeton University
Princeton NJ 08544.

Abstract

This paper treats a signal detection problem using arctic under-ice noise. The authors have had access to one large segment of data (6150144 samples), which is nonstationary and has been shown to be non-Gaussian. A model is presented for the arctic under-ice noise, and the performance levels achieved by several different detectors are compared. The association between the shape of the empirical probability density function and the shape of the power spectrum is explored. The arctic noise is known to contain narrowband and impulsive components, and it is shown here that removal of the narrowband components results in nearly Gaussian noise.

ARCTIC UNDER-ICE NOISE

In the early 1940's, researchers began examining the characteristics of "open-water" noise [1,2], but it was not until the late 1950's that an interest developed in underwater acoustics in ice-covered areas. The main objectives were to determine average pressure spectra levels (hereafter called noise levels) as a function of weather conditions, and to gain an understanding of the physical processes causing under-ice noise. Under-ice noise levels, for shore-fast and floe ice, were shown to be unaffected by wind speed and to be anisotropic. Several researchers concluded that the majority of under-ice noise was the result of slight movements of the ice itself and temperature gradients causing the ice to crack. They also found that these physical processes resulted in non-Gaussian and impulsive noise, with the noise becoming more Gaussian in nature with increasing depth [3-6].

By the late 1960's, researchers began to concentrate on single records to obtain statistical information about the under-ice environment. This paper involves such an analysis performed on FRAM II data. The data was recorded on 23-24 April 1980, from a pack ice floe, at $86^{\circ} N. 25^{\circ} W.$. An omnidirectional hydrophone, radio linked to a receiver, was suspended to a depth of 91m in 4000m deep water. The data, recorded on an analog device, was bandpass filtered from 0.01-5kHz. Next the data was passed through a lowpass filter with a 2.5kHz cutoff, then it was digitized with a sampling rate of 10kHz. The specific data segment being analyzed was recorded from 11:30 - 11:40 pm on April 23. This segment contains a total of 6,150,144 samples, which is slightly longer than 10 minutes [7,8]. For convenience, it has been broken up into 6006 records of 1024 numbers each.

A number of authors have looked at this data segment. Dwyer [7,9,10] took several different approaches to analyzing the under-ice data. He examined the time-domain data and concluded from an analysis of the first four central moments that the noise is nonstationary and non-Gaussian. The non-Gaussian nature was confirmed when a time-domain energy spectrum was compared to that for a Gaussian noise source. The same techniques enabled the identification of narrowband and impulsive components. The existence of these components was not surprising, since the floe was moving slowly throughout the experiment [6,9]. A similar analysis carried out in the frequency domain yielded comparable results.

Veitch and Wilks [8] attempted to develop a model for the under-ice noise. Their work confirms the presence of narrowband and impulsive components, but suggests that the background noise is Gaussian. By examining the first four central moments of the time-domain data in detail, they noted a strong correlation between records with large kurtosis and large skew. They also determined that a large kurtosis was associated with high amplitude bursts. They proposed that the under-ice noise could be modeled as a mixture containing a Gaussian background, high powered sinusoids of random frequency and amplitude, and an impulsive component. We agree that this model is valid, though a specific model for the impulsive component is not presented [see [8] for a detailed discussion of the model].

Since the noise was known to be nonstationary, we began an analysis aimed at estimating the rate at which the statistical characteristics of the noise were changing. Empirical probability density function (epdf) plots were generated, each based on 5 contiguous records ($\approx$ 0.5 sec). Careful examination of the epdf's indicated that the noise could be modeled by a member of the generalized Gaussian family. The generalized Gaussian families of densities can be represented in the form:

$$f(x) = Be^{-(b\,|x-\mu|)^c}$$

where

$$b = \frac{1}{\sigma}\left[\frac{\Gamma(3/c)}{\Gamma(1/c)}\right]^{1/2} \qquad B = \frac{bc}{2\Gamma(1/c)}$$

where $\mu =$ mean, $\sigma^2 =$ variance, $c =$ the shape parameter, and $\Gamma(\cdot)$ is the gamma function. The other main parameters of interest for this family are the skew and the kurtosis. The skew is zero for all values of c, and the kurtosis can be expressed as:

$$Kurtosis = \frac{\Gamma(1/c)\,\Gamma(5/c)}{\Gamma(3/c)^2}$$

It can be easily shown that the kurtosis can take on values as high as 6 when c=1, and as low as 2 when c=6. For $c\epsilon(1,6]$, the relationship between kurtosis and c is one-to-one. A discussion on the usefulness of kurtosis as a measure can be found in [8,11]. A simple time-varying detector structure was implemented where the mean and variance were estimated by the sample mean and sample variance, and the sample kurtosis was used to estimate c. For $c \in (1,6]$, results show that this family models the epdf well if the epdf is nearly symmetric and the kurtosis is less than 4.

For the simple detection problem, we have:

$$H\colon\ X_i = N_i$$

$$K\colon\ X_i = N_i + s$$

where X_i is the received signal, N_i is the arctic under-ice noise, and s is a positive known signal. Define the *false alarm probability*, α as:

$$\alpha = P(\ decide\ K\ |H\ is\ true)$$

and the *power*, β as:

$$\beta = P(\text{ decide } K \mid K \text{ is true})$$

Define the *performance level* as the value of β for a given α achieved by a detector, with the goal to find the detector which maximizes the performance level. In order to judge the effectiveness of the generalized Gaussian model for the under-ice noise, the performance levels of the following detectors was compared.

[D1] The Linear Detector (Neyman-Pearson optimal for Gaussian noise)

[D2] The Amplitude Limiter Detector (Neyman-Pearson optimal for Laplace noise)

[D3] The Time-Varying Linear Detector

[D4] The Time-Varying Generalized Gaussian Detector

To determine the power levels achieved by detector structures (D1) and (D2), it was assumed that the global mean and variance of the noise were known to the detector. From this information, the signal power was determined in order to achieve an overall S/N ratio of unity. The entire data set was then used as an input to the detector in order to calculate the epdf for the detector structure, from which α and β are easily determined. A different approach was used with (D3) and (D4). For these two detectors, the (α,β) pairs were computed for blocks of 5 records (blocks of 5 records were chosen to insure that the data is as stationary as possible within each block), with the mean, variance, and kurtosis of each block known to the detector, which then adjusted the structure and the signal power in order to achieve a S/N ratio of unity. It is emphasized that the structures of (D1) and (D2) are fixed for the *entire* data set, while the structures of (D3) and (D4) change with every record.

Figure 1 illustrates the (α,β) pairs for (D1) - (D4), where the average levels of β for each α for (D3) and (D4) are plotted. (D2) is optimal in the Neyman-Pearson sense for Laplace noise, and while the presence of high amplitude samples in the noise have been confirmed, the samples generally occur in bursts lasting a very short time. The arctic data also has Gaussian characteristics much of the time. Combining the above, we would expect that (D1) would be better than (D2). Our results agree with this hypothesis. Due to the nonstationary nature of the noise, the time-varying detectors were both expected to outperform the linear detector. As predicted, the performance levels of (D3) and (D4) were superior to (D1). The figure also indicates that there is essentially no difference in performance levels for (D3) and (D4).

A more informative comparison between these two detectors is displayed in Figure 2. This is a plot of the ratio of power levels achieved for $\alpha = 0.10$ on a block-by-block basis. Note that the mean of β_3/β_4 is almost unity and the variance is very small. For the generalized Gaussian density written in the above form with zero mean, the Neyman-Pearson optimal nonlinearity $t(x)$ has the form:

$$t(x) = -b^c \, |x-s\,|^c + b^c \, |x\,|^c$$

As c varies around the value of 2, $t(x)$ becomes nonlinear, but the change is small and slow, so that if c is near 2, then $t(x)$ is nearly linear. In fact, for this data, the average value of c that (D4) was adjusted to was 2.16, with $c \in [1.75,2.25]$ for most blocks. This implies, that based on the sample kurtosis, the detector thought the arctic data to be slightly lighter tailed than Gaussian. Given this information, the similar performance levels of (D3) and (D4) are understandable.

In order to gain an understanding of the frequency content of the arctic data, power spectra were computed for 0.5 second blocks of data, from the beginning to the end of the data set. In agreement with [7,8], high powered tonals which change center frequency and amplitude are apparent throughout the data set, where the tones are normally in the frequency band 0.8 - 1.8 kHz, with resonances often at higher frequencies. In an effort to understand how these tonals affect the statistics of the arctic noise both epdf's and power spectra were computed, for blocks of 5 records. Unexpectedly, there appears to be an association between the overall shape of the power spectrum and the shape of the epdf.

Specifically, for blocks whose power spectra show high-powered tonals, the epdf differs markedly from Gaussian - principally in becoming light-tailed. Recall that adding a sinusoid (of random phase and amplitude) to Gaussian noise will result in lighter tails than Gaussian, and that [8] proposed the arctic noise could be modeled as a mixture which includes a Gaussian background plus added sinusoids of random amplitude and phase. Figure 3 is an example of this phenomenon. The periodogram, 3(a), shows a high powered tonal centered at 1125 Hz with what appears to be a resonance at 2250 Hz. In 3(b), the epdf for the same data, and a Gaussian curve are plotted, with the tails enlarged in 3(c). The epdf for this block has heavier shoulders, flatter center, and lighter tails than Gaussian. For this particular block, (D4) had a slightly better performance level than (D3).

For blocks whose power spectra show moderate amplitude resonances, typically the epdf is more peaked and heavier-tailed than Gaussian. An example is shown in Figure 4. Note in particular, that the data has heavier tails than Gaussian, as illustrated in 4(c). For the records used in Figure 4, we find (D3) is marginally better than (D4). This could be due to the limited ability of our detector to model records which have a large kurtosis. Finally, for blocks whose power spectra appear to have no distinguishing characteristics (no visible resonances or spikes), the epdf resembles a Gaussian curve.

Now that we had some understanding of the association between the power spectrum and epdf for a data block, we decided to use this knowledge in the hope of gaining a better understanding of the statistics of the arctic under-ice noise. The obvious place to begin is with the blocks which have high powered tonals in their power spectra. We decided to preprocess the data, with the goal that the processed data would be Gaussian in nature.

A common technique used in signal processing to separate a sinusoidal signal of unknown frequency from background noise is to use an adaptive notch filter (ANF). The ANF is either recursive or non-recursive, and based on the data adjusts parameters to move the center frequency of the notch. Generally, there is also a parameter to adjust the width of the notch, which may be adaptive or fixed. This technique was used on blocks which had a high powered tonal in their power spectrum. The particular filter used was [1]:

$$H(z) = \frac{1+bz^{-1}+z^{-2}}{1+\gamma bz^{-1}+\gamma^2 z^{-2}}$$

where b determines the center frequency of the notch and γ determines the width of the notch. Generally, $\gamma \in [0.8, 0.95]$ for best results. The actual value of b is found by a simple gradient search algorithm and is the only adaptive parameter for the filter. The idea was to remove as much of the power of the tones as possible and then determine if the resulting background noise was Gaussian. In contrast to the ideal nonlinearity suggested by [10], we used time domain processing.

As an example, records 4101-4105 were processed in this fashion. The notch width γ was fixed at the value of 0.85. Increasing γ resulted in removing too little of the power of the narrowband component, while decreasing γ resulted in removing too much power not associated with the narrowband component. The spectrum of the filtered data is shown in Figure 5. Note that the narrowband component centered at 1125 Hz has a significantly lower power, though it was not removed completely, and that the resonance appears not to be affected by the filtering. To determine what effect the reduction of the tonal power had on the shape of the epdf, the filtered data was then used to compute a new epdf. Plotted in Figure 6, are the original and filtered epdf's, along with a Gaussian curve of mean zero and variance unity. The filtered data has lighter shoulders and heavier tails than the unfiltered data, and the epdf compares well with the Gaussian curve. Processing other blocks which had a high powered tonal yielded similar results.

A comparison was made between the performance levels of the filtered and raw data for these same records. The time varying linear detector (D3) was used for this purpose. Figure 7 is a plot of Figure 1 but including the (α,β) curves for the raw and filtered data. First note that due to the asymmetry of the epdf's, the power curve is not completely symmetric. Also recall that the performance levels for (D3) and (D4) are *average* power levels for each α. We see that filtering the data does result in an increase in the power level. Note also that the power level for the filtered data compares well with the average power achieved by (D3).

SUMMARY

While the generalized Gaussian model seems to fit the empirical probability density function plots of the arctic data fairly well, there is no detection advantage apparent. This suggests that, for purposes of detection, there is no reason to increase the complexity to use this model. It is also clear, that at the very least a time-varying detector structure should be used, and most likely an adaptive structure would yield the best results.

It was shown that, for blocks which contain a high powered tonal, an adaptive notch filter can be used to preprocess the data, which results in more nearly Gaussian data and a performance improvement. There are several problems with using such a structure however. If one uses a sinusoidal carrier, then precautions must be taken to insure the filter does not remove the signal. Care must also be taken to insure that the filter can handle multiple narrowband components, and that it becomes all-pass when there exists no narrowband component. The adaptive notch filter does not address the problem suggested in Figure 4. That is, the ANF is not a proper preprocessing method for data which is heavy tailed relative to Gaussian.

ACKNOWLEDGEMENT

This work was supported by the Office of Naval Research under Contract N00014-81-K-0146 and by the National Science Foundation under Grant ECS-8713598.

REFERENCES

[1] P. Willett, *Density Representations with Applications to Signal Detection* , PhD Dissertation, Princeton, 1986.

[2] G.M. Wenz, "Review of Underwater Acoustics Research: Noise," *Journal of the Acoustical Society of America* 51 (3), pp. 1010-1024, 1972.

[3] A.R. Milne, "Shallow Water Under-Ice Acoustics in Barrow Strait," *Journal of the Acoustical Society of America* 32 (8), pp. 1007-1016, 1960.

[4] J.D. Macpherson, "Some Under-Ice Acoustic Ambient Noise Measurements," *Journal of the Acoustical Society of America* 34 (8), pp. 1149-1150, 1962.

[5] C.R. Greene and B.M. Buck, "Arctic Ocean Ambient Noise," *Journal of the Acoustical Society of America* 36, pp. 1218-1220, 1964.

[6] A.R. Milne and J.H. Ganton, "Ambient Noise Under Arctic-Sea Ice, " *Journal of the Acoustical Society of America* 36 (5), pp. 855-863, 1964.

[7] R. Dwyer, "Arctic Ambient Noise Statistical Measurement Results and their Implications to Sonar Performance Improvements," in *Proceedings of Conference on Undersea Ambient Noise* (SACLANT ASW Research Centre, La Spezia, Italy), 11-14 May 1982.

[8] J.G. Veitch and A.R. Wilks,"A Characterization of Arctic Undersea Noise," *Journal of the Acoustical Society of America* 77, pp. 989-998, 1985.

[9] R.F. Dwyer, "A Technique for Improving Detection and Estimation of Signals Contaminated by Under Ice Noise," *Journal of the Acoustical Society of America* 74 (1), pp. 124-130, 1983.

[10] R.F. Dwyer, "Essential Limitations to Signal Detection and Estimation: An Application to the Arctic Under Ice Environmental Noise Problem," *Proc IEEE* Vol. 72 No. 11, pp. 1657-1660, 1984.

[11] R.S. Burington and D.C. May, *Handbook of Probability and Statistics with Tables,* Handbook Publishers, Inc., Sandusky, Ohio, 1953.

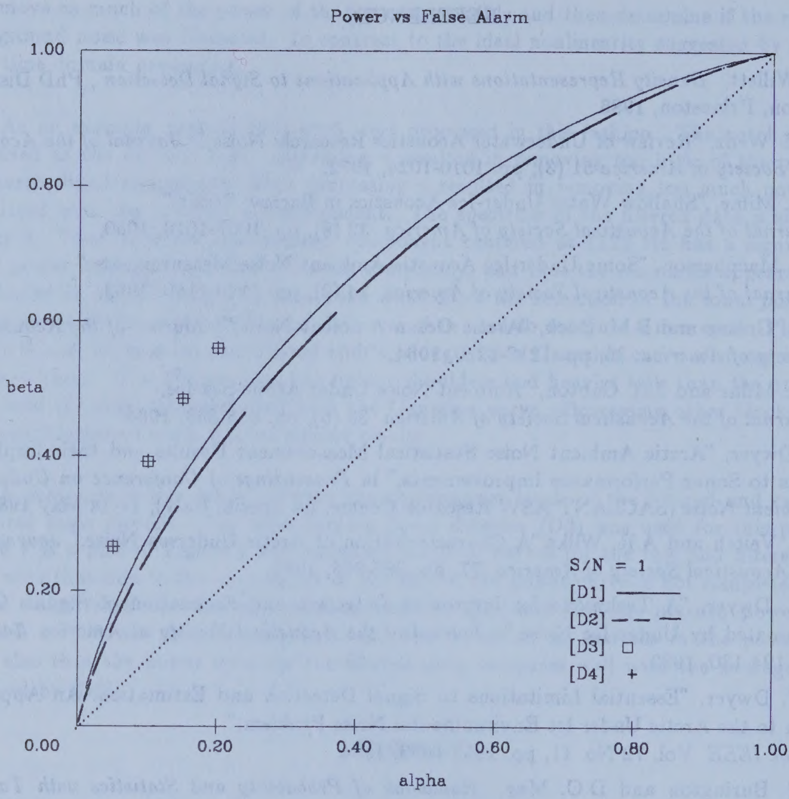


Figure 1

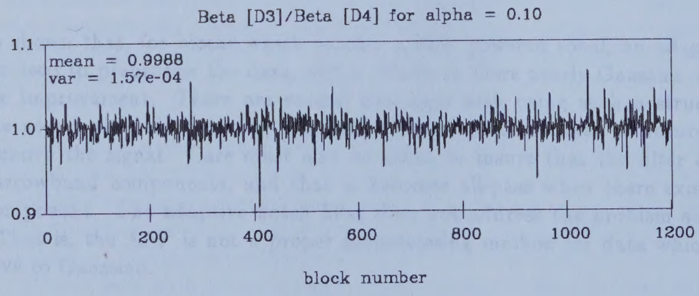
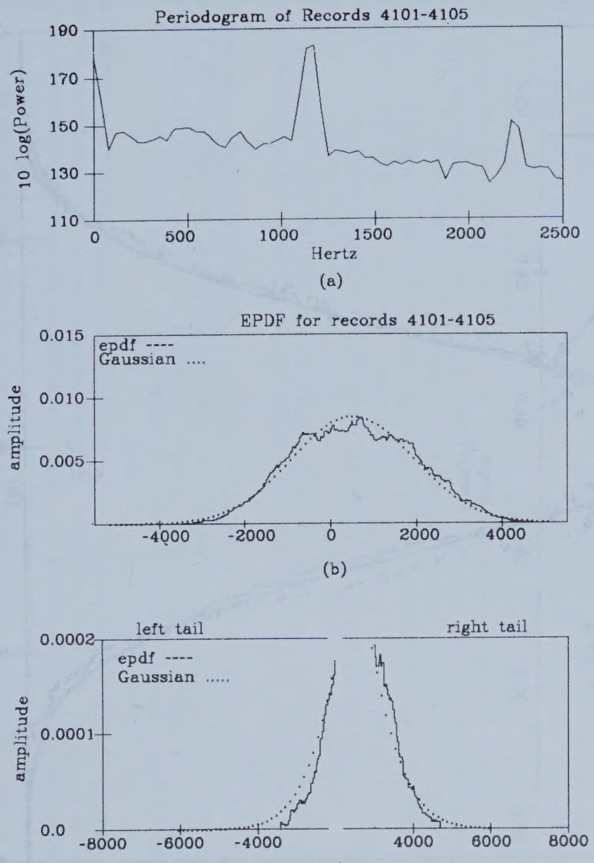


Figure 2

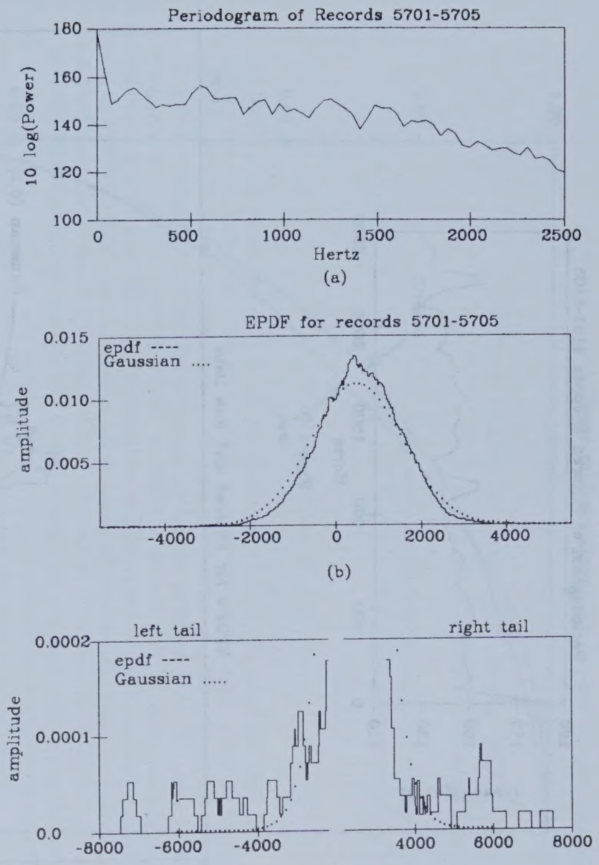


(a)

(b)

(c)

Figure 3



(a)

(b)

(c)

Figure 4

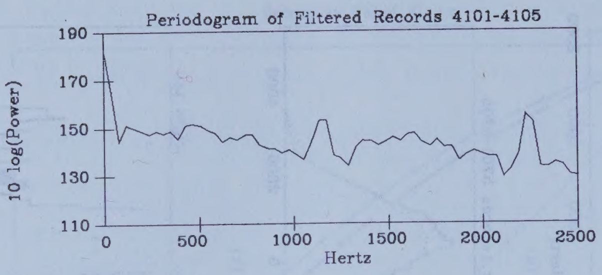


Figure 5

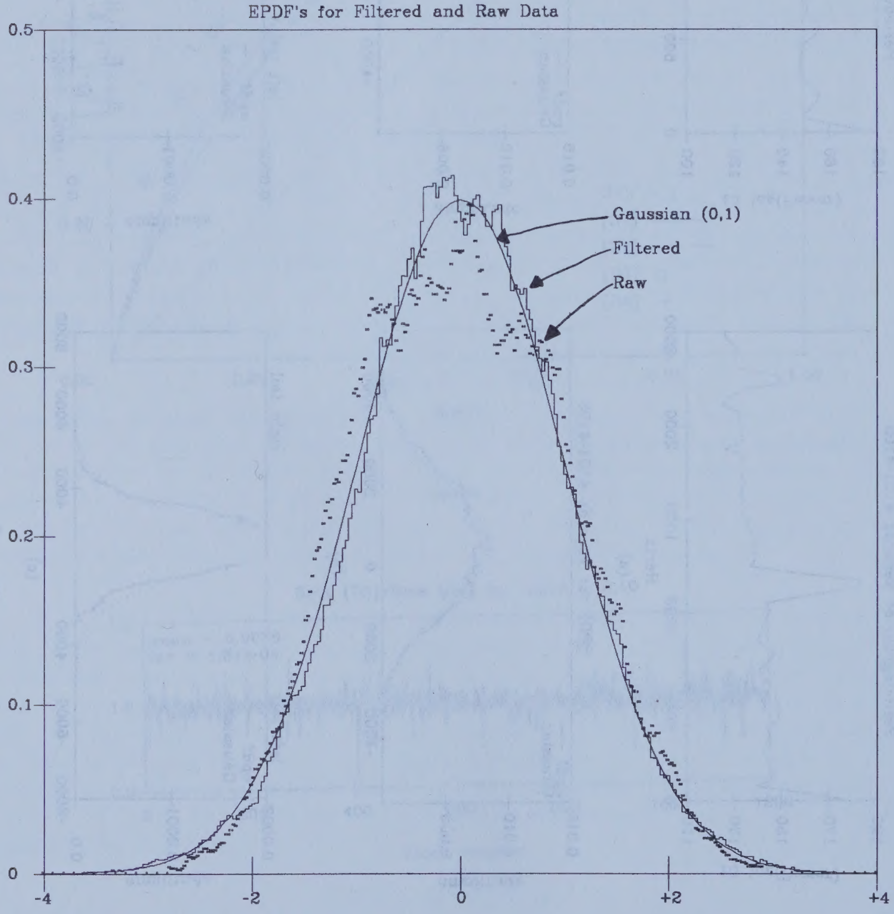


Figure 6

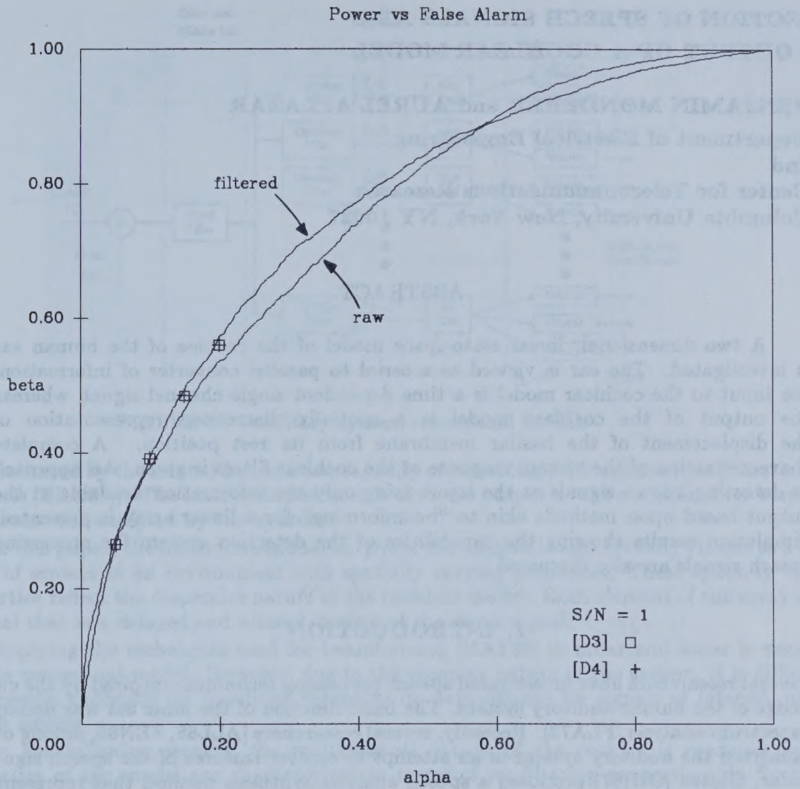

Power vs False Alarm
1.00
0.80
0.60
0.40
0.20
0.00
beta
filtered
raw
S/N = 1
[D3] □
[D4] +
0.20
0.40
0.60
0.80
1.00
alpha

Figure 7

DETECTION OF SPEECH SIGNALS AT THE OUTPUT OF A COCHLEAR MODEL

BENJAMIN MONDERER and AUREL A. LAZAR
Department of Electrical Engineering
and
Center for Telecommunications Research
Columbia University, New York, NY 10027

ABSTRACT

A two dimensional, linear state-space model of the cochlea of the human ear is investigated. The ear is viewed as a serial to parallel converter of information: the input to the cochlear model is a time dependent single channel signal, whereas the output of the cochlear model is a spatially discretized representation of the displacement of the basilar membrane from its rest position. A complete characterization of the system response of the cochlear filters is given. An approach to detecting known signals at the input using only the information available at the output based upon methods akin to "beamforming" for a linear array is presented. Simulation results showing the capabilities of the detection system for processing speech signals are also discussed.

1. INTRODUCTION

Several researchers have investigated speech processing techniques inspired by the current knowledge of the human auditory system. The basic function of the inner ear was understood to be spectrum analysis [FLA72]. Recently, several researchers [ALL85, SEN85, among others] have analyzed the auditory system in an attempt to recover features of the speech signal. In particular, Ghitza [GHI87] proposed a speech analysis/synthesis method that represents the speech spectrum as the output process of an auditory system model. The conjecture that the cochlea acts as a delay line has also been [ROB85, YEG85].

The auditory system is viewed as a *serial-to-parallel* converter of information, as shown in Figure 1 . In the serial-to-parallel paradigm, acoustic waves are filtered by the outer and middle ears, and subsequently enter the cochlea longitudinally through the oval window of the cochlea. These waves are transformed by the cochlea into a transversal motion of the basilar membrane. The information contained in the displacement of the basilar membrane is then transmitted as neural information to the brain in the form of a multichannel neural spike train (point process).

For steady-state signals, the resolution capabilities of a detection system at the output of the cochlea has been shown to outperform a detection system at the single channel input [MON87]. The resolution performance measure employed was suggested by the ***ambiguity function*** [RIH69]. The ambiguity function is equivalent to the Wigner Distribution previously used for analyzing time-varying signals [BOU86].

As a filter bank, the function of the basilar membrane appears similar to the one of a spectrum analyzer. The interpretation of the cochlea as a spectrum analyzer goes back to Helmholtz [HEL77] in the last century. This interpretation remains the predominant model for inner ear function [GHI87].

The use of filter banks in speech processing is common in sub-band coding systems. In these systems, a bank of bandpass filters is used to break up the input signal into frequency bands. The output of each of these filters can be sampled at a rate significantly less than the Nyquist rate for the input signal. Recent results [VET87] clarify the design of filter banks in sub-band coding systems that achieve complete signal recovery.

The cochlear filter bank acts as a mechanism for converting the wideband signal ($\approx$ 20,000 kHz.) at the input into a set of outputs with much smaller bandwidths. This smaller bandwidth

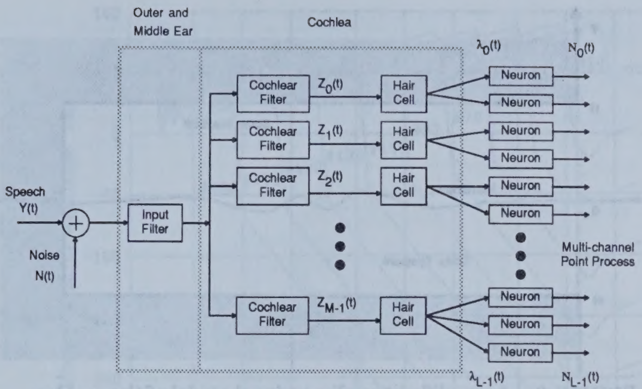


FIGURE 1. Auditory system conceptual model.

is necessitated by the limited information capacity of individual auditory neurons (point process channels [BRE81]). The interpretation here is that the output of the cochlear filter bank is in some sense sub-sampled by the neurons.

In this paper, the multidimensional output of the cochlea is conceptually viewed as a linear array of sensors in an environment with spatially varying properties. These spatially varying properties reflect the dispersive nature of the cochlear model. Each element of the array senses a signal that is a delayed and altered version of the input signal.

Applying the techniques used for beamforming [HAY85] in radar and sonar is suggested by this conceptual model. However, due to the complex nature of the system, it is difficult to gain results explicitly for questions regarding its performance for signals of interest, namely speech signals.

In the following section, the multichannel output of the cochlea is modeled and the properties of the model are characterized in terms of its filtering properties. In Section 3, the filter bank model is used for examining the detection problem at the output of the cochlea. In Section 4, the space-time filters from the detection system are applied to speech signals. Finally, the conclusion examines what has been accomplished and suggests some possibilities for future work.

2. COCHLEAR MODEL AND CHARACTERIZATION

Analysis of the motion of the basilar membrane in a two-dimensional fluid mechanical model of the cochlea, as in Figure 2 leads the integro-differential equation [MON86]

$$4\rho\int_0^L g(x,y)\frac{\partial^2 Z(y,t)}{\partial t^2}dy + 2(L-x)\rho\frac{\partial Y(t)}{\partial t} = - k(x)Z(x,t) - b(x)\frac{\partial Z(x,t)}{\partial t} - m(x)\frac{\partial^2 Z(x,t)}{\partial t^2}\ , \tag{1}$$

for all x, $x \in [0, L]$, where $Y(t)$ is the displacement of the oval window (input), $Z(x,t)$ is the displacement of the basilar membrane at position x and time t (output), $k(x)$, $b(x)$ and $m(x)$ are the stiffness, damping and mass of the membrane, ρ is the density of the fluid in the cochlea, $g(x,y)$ is the Green's function of the Laplace equation for the given domain, L is the length of the cochlea. The initial conditions are: $t = 0$, $Z(x,0) = 0$ and $\partial Z(x,t)/\partial t|_{t=0} = 0$ for all $x \in [0, L]$. Simulation of this equation in continuous time and discrete time is accomplished by discretizing the spatial variable. From this formulation, the state-space model of the cochlea is developed [MON87],

$$\ddot{\mathbf{Z}}(t) = \mathsf{D}^{-1}\left(-\mathsf{B}\dot{\mathbf{Z}}(t) - \mathsf{C}\mathbf{Z}(t) + \mathsf{d}\dot{Y}(t)\right)\ , \tag{2}$$

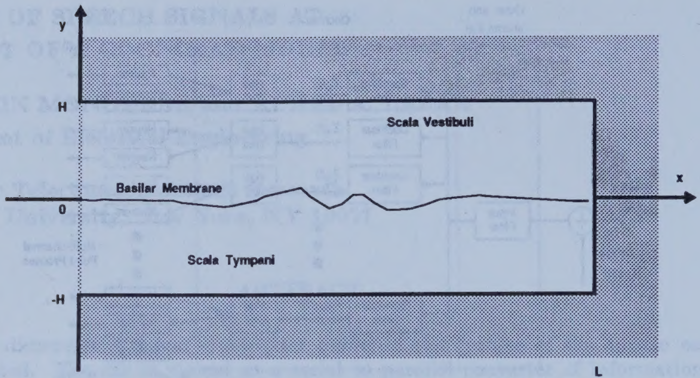


FIGURE 2. A simplified two-dimensional model of the cochlea [MON86].

where $\mathbf{Z}(t)$ is a vector of M elements. B, C and D are $M \times M$ matrices and d is a M vector all of which are derived from the parameters of the cochlear model given above.

The state space model of the displacement of the basilar membrane is considered to be a bank of filters in which each filter's output represents the displacement of the membrane at a particular point in space. The transfer function from the input to each channel of the output is derived using standard methods for linear systems [KAI80]. Let

$$\mathsf{A} = \begin{pmatrix} -\mathsf{D}^{-1}\mathsf{B} & -\mathsf{D}^{-1}\mathsf{C} \\ \mathsf{I} & 0 \end{pmatrix} \quad \mathbf{q} = \begin{pmatrix} \mathsf{D}^{-1}\mathbf{d} \\ 0 \end{pmatrix} \quad \mathsf{E} = (\,0 \quad \mathsf{I}\,)\,, \tag{3}$$

where A is a $2M \times 2M$ matrix, q is a $2M$ vector and E is a $2M \times M$ matrix.

The solution to equation (2) is

$$\mathbf{Z}(t) = \mathsf{E}\int_0^t e^{\mathsf{A}(t-\tau)}\mathbf{q}\dot{Y}(\tau)d\tau\,, \tag{4}$$

where $e^{\mathsf{A}(t-\tau)}$ is the matrix exponential. Applying the Laplace transform to equation (4) , we have $\mathbf{Z}(s) = \mathbf{H}(s)Y(s)$, where $\mathbf{H}(s)$ is the transfer function from $Y(s)$ to $\mathbf{Z}(s)$ and

$$\mathbf{H}(s) = \mathsf{E}(s\mathsf{I} - \mathsf{A})^{-1}\mathbf{q}s\,. \tag{5}$$

The i-th filter transfer function $H_i(s)$ is given by

$$H_i(s) = \frac{\mathbf{e}_i \mathrm{adj}(sI - \mathsf{A})\mathbf{q}s}{\det(sI - \mathsf{A})}\,, \tag{6}$$

and therefore, is the ratio of two polynomials. Notice that the poles of all of the filters are the same; the difference between the filters is determined completely by the zeroes.

The poles and zeroes, the impulse response, and the frequency response for the filter banks have been calculated. Plots of the poles and zeroes, the frequency responses of the filters for a particular set of model parameters are shown in Figure 3 .

Figure 3 shows the frequency responses of several of the filters for $M = 200$. Figure 4 shows the poles for filter # 100 and Figure 5 shows the zeroes on the same scale. The configuration of the zeroes is typical of all of the filters. There is a group of zeroes that appear far from the main grouping and a set of zeroes that form "umbrellas" symmetric about the real axis.

In the cochlear modeling literature, questions have been raised as to the origin of resonances in the linear model of the cochlea [MAT80], and the origin of the hump in the frequency response at the high frequencies of every filter [NEE81]. The resonances at the low frequencies is due to the spatial discretization which results in a finite number of poles and zeroes. The resonant frequencies move around with different values of M. In Figure 6 , the poles (+) are very close to the imaginary axis and are not canceled out by nearby zeroes (O). The hump at the high frequencies in Figure 3 is due to the configuration of the poles and zeroes far away from the origin, see Figure 6 and 5 .

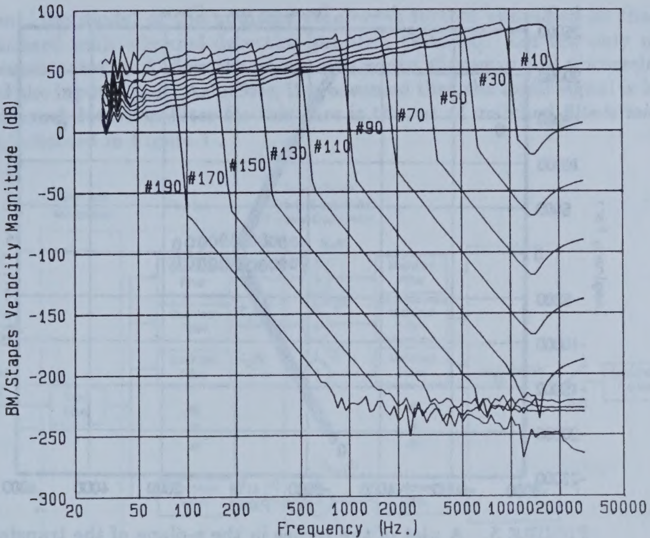


FIGURE 3. Frequency response of 10 filters for $M = 200$.

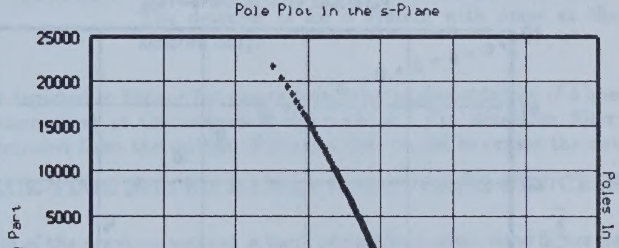


FIGURE 4. A plot of the poles in the s-plane of the transfer function $H_{100}(s)$.

3. DETECTION AND FILTERING

In our model, the sensors on the membrane are considered to form an array. Processing the outputs of this sensor array is similar to "beamforming" used by radar and sonar [HAY85]. The detection problem of interest is one in which a known signal is detected to be present or absent. Example signals might be speech or sinewaves.

The model of the system is shown in Figure 7 . This model consists of an input signal $y(t)$ which may be corrupted by interfering noise that enters the ear. The signal and noise are

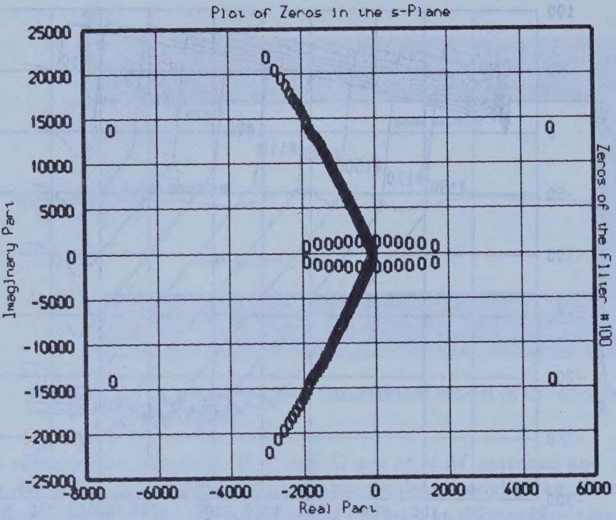


FIGURE 5. A plot of the zeroes in the s-plane of the transfer function $H_{100}(s)$.

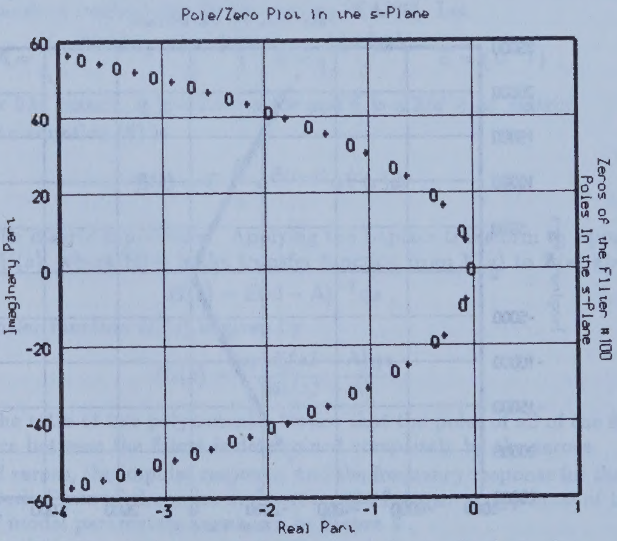


FIGURE 6. A close-up of the zeroes and poles in the s-plane of the transfer function $H_{100}(s)$.

filtered by the cochlear filter bank to create a vector of outputs,

$$\mathbf{Z}(t) = [Z_0(t), \ldots, Z_{M-1}(t)] \, , \tag{7}$$

where M is the number of outputs of the cochlear filter bank. Each output of the filter bank is corrupted by "sensor" noise. This noise represents "sampling noise" which occurs as a result of the conversion of the basilar membrane motion to the hair cell potentials and subsequently to the neural spike train. The vector of outputs is then processed by a filter that produces a single channel waveform. This waveform is a likelihood ratio that is passed through a threshold detector to detect the presence of the signal.

At this point, the model of the auditory receiver is further simplified so that it can be related to a standard multi-channel detection problem [VAN68]. Let the only noise in the system be the sensor noise, and allow the noise to be white, Gaussian and uncorrelated among the channels and the input signal. As before, it is assumed that the input signal is known. The optimal multi-channel detection filter for this case is the set of matched filters feeding into a beamformer, as indicated in Figure 7 .

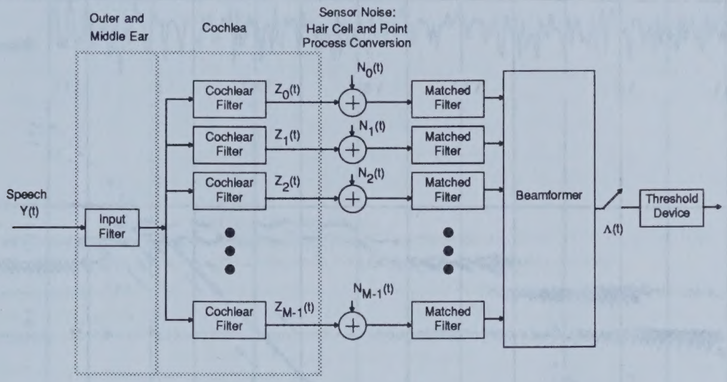


FIGURE 7. Diagram of the model of the "auditory receiver." The detector is for a system with noise at the sensors only.

The system depicted in Figure 7 suggests a parallel implementation of a speech processing system using information at the output of the cochlea. The detection filter combines the space-time information from the output of the cochlear model to create the detector.

4. SPACE-TIME FILTER BANKS FOR SPEECH PROCESSING

Using results of the previous section, a bank of matched space-time filters has been created to observe the output of the cochlear model. The output of each filter in the bank represents the likelihood ratio for a prototype signal. Choosing signals for the matched filters determines the properties of the filter bank.

Two types of the results of simulation experiments with transient speech signals are presented below. The first shows a spectrogram-like display of the input signal by using narrowband filters created from the steady-state response of the basilar membrane for a set of sinewaves. The second shows the use of a bank of filters tuned to detect specific signals of a known speech input.

The filters for these experiments were created by taking a "snap-shot" of the basilar membrane displacement and using it for the coefficients of the beamformer. The spatial filter is "matched" with the cochlear output for a single observation in time. This approach simplifies the computations while maintaining the basic properties of the system.

Figure 8 shows an example of speech processed using the narrowband filters. The signal is a continuous sequence of vowels uttered by a male adult maintaining almost constant pitch. The vertical axis corresponds to the bank of detection filters arranged according to frequency. The horizontal axis represents time. The output of each filter is represented by the intensity at each point in time and space. This figure reads similarly to a standard speech spectrogram. The difference is that this diagram has a logarithmic distribution of filters as opposed to a linear distribution. Figure 9 shows the magnitude of the frequency response of 10 of the filter transfer functions. The beamformer coefficients for the filter can be found in [MON87].

A set of snapshots of the basilar membrane response to a speech signal is used to define the filter bank. Figure 10 shows an example of a speech signal processed using the speech detection filters. This figure is read similarly to Figure 7 , except that the filters are labeled

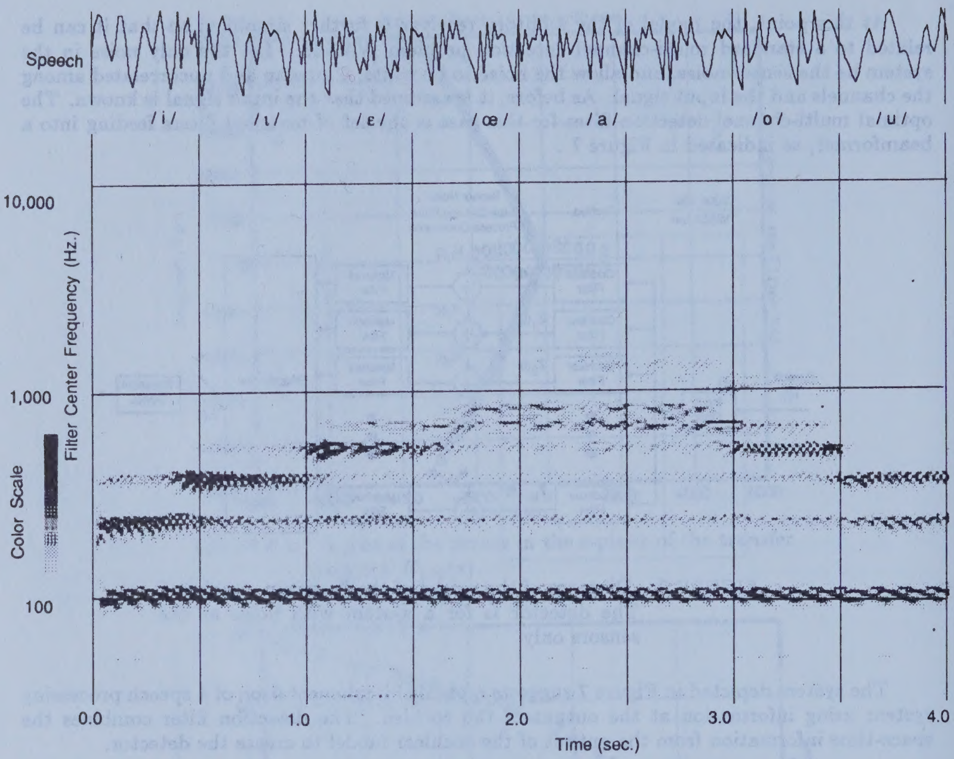


FIGURE 8. Output of the spectrogram-like filter bank.

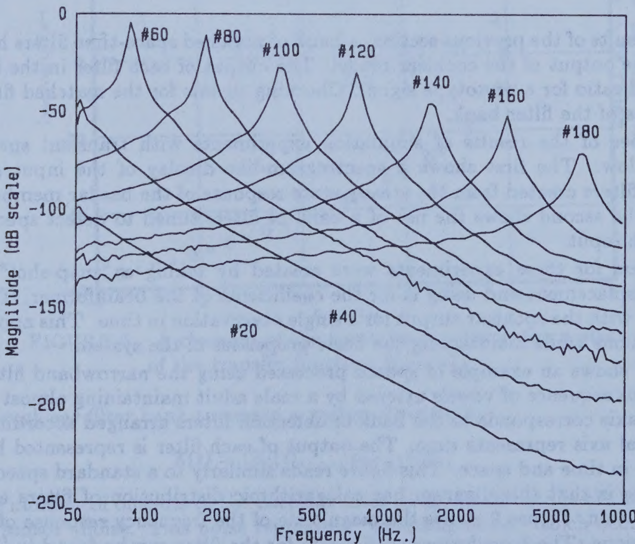


FIGURE 9. Frequency response of 10 of the beamformer filters from the spectrogram-like filter bank.

with the portion of the speech that corresponds to the snap shot. The resolution of the filters is narrow as evident by the narrow diagonal running from top left to bottom right. This diagonal is where the filter matches the input signal. Notice that the response of similar vowel detection filters respond to each occurrence of the vowel in the sentence, especially at the bottom during the initial and final vowels.

The beamformer coefficients and the magnitude of the frequency response of filter # 77 are shown in Figures 11 and 12 , respectively.

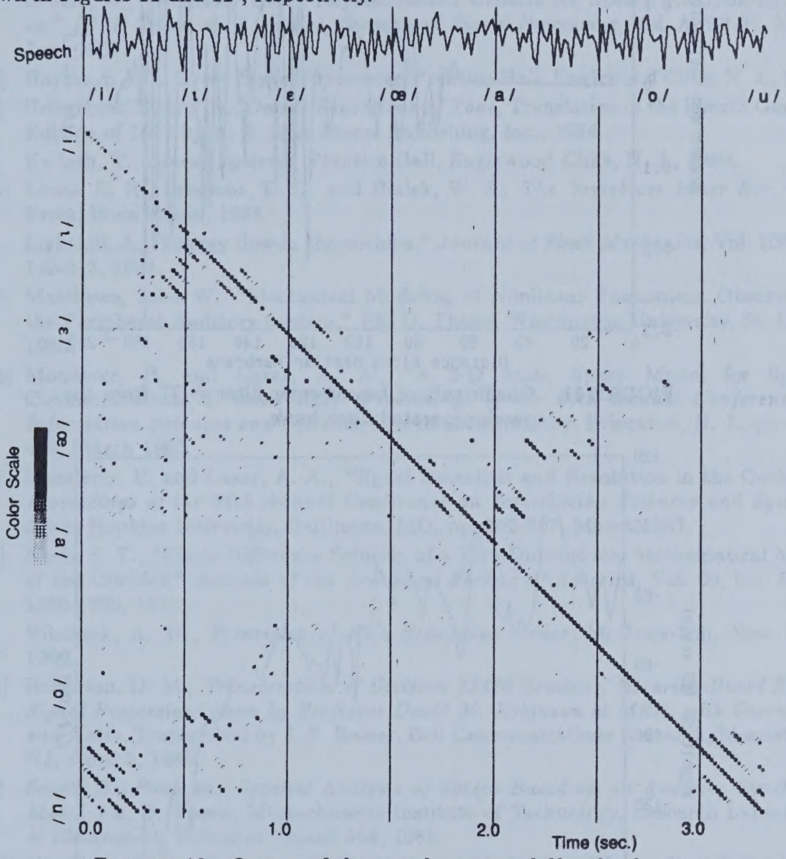


FIGURE 10. Output of the speech generated filter bank.

5. CONCLUSION

A two-dimensional cochlear model has been analyzed using linear system theory. The space-time output of the cochlear model has been used to develop a system for processing acoustic information. The basic characteristics of the latter system are based on results on multi-channel detection theory.

Note that it is very simple to create filters for the filter bank from prototypes of the basilar membrane motion. All that is necessary for the implementation is the weighting coefficients for each channel and a summing element.

Further work needs to be done to find the properties and uses of this type of system for various speech signals and natural sounds. In particular, the resolving capabilities of the system in the presence of interfering environmental signals should be investigated. A system such as the one described might be used in speech processing with models of the auditory

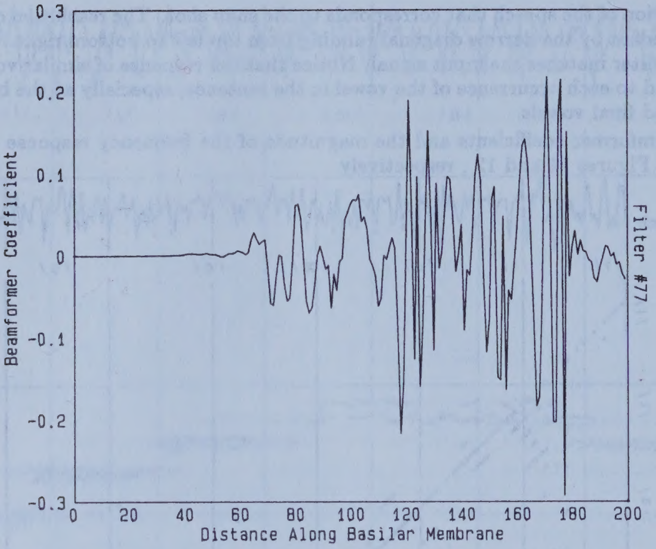


FIGURE 11. Coefficients of beamformer filter # 77 from the speech generated filter bank.

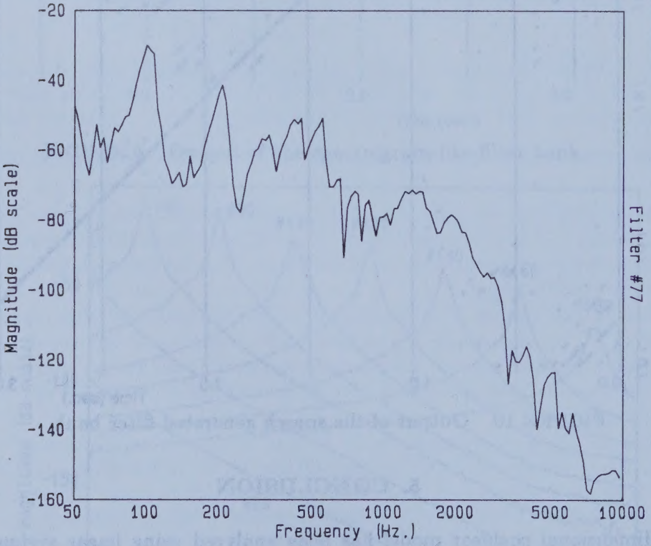


FIGURE 12. Frequency response of beamformer filter # 77 from the speech generated filter bank.

neural system. Finally, an interesting question is how can the parameters of a voice signal be identified using only information at the periphery of the auditory system.

The work reported here was supported by the National Science Foundation under Grant CDR-44-21402.

REFERENCES

[ALL85] Allen, J. B., "Cochlear Modeling," *ASSP Magazine,* Vol. 2, No. 1, Jan. 1985, pp.

3-29.

[BOU86] Boudreaux-Bartels, G. F. and Parks, T. W., "Time-Varying Filtering and Signal Estimation Using Wigner Distribution Synthesis Techniques," *IEEE Trans. on Acoustics, Speech and Signal Processing,* Vol. ASSP-34, No. 3, June 1986.

[BRE81] Brémaud, Pierre, *Point Processes and Queues, Martingale Dynamics,* Springer-Verlag, NY, 1981.

[GHI87] Ghitza, O., "Auditory Nerve Representation Criteria for Speech Analysis/Synthesis," *IEEE Trans. on Acoustics, Speech and Signal Processing,* Vol. ASSP-35, No. 6, June 1987.

[HAY85] Haykin, S., ed., *Array Signal Processing,* Prentice-Hall, Englewood Cliffs, N. J., 1985.

[HEL77] Helmholtz, H. L. F. v., *On the Sensations of Tone,* Translation of the Fourth German Edition of 1877 by A. J. Ellis, Dover Publishing, Inc., 1954.

[KAI80] Kailath, T., *Linear Systems,* Prentice-Hall, Englewood Cliffs, N. J., 1980.

[LEW85] Lewis, E. R., Leverenz, E. L., and Bialek, W. S., *The Vertebrate Inner Ear,* CRC Press, Boca Raton, 1985.

[LIG81] Lighthill, J., "Energy flow in the cochlea," *Journal of Fluid Mechanics,* Vol. 106, pp. 149-213, 1981.

[MAT80] Matthews, John W., "Mechanical Modeling of Nonlinear Phenomena Observed in the Peripheral Auditory System," Ph. D. Thesis, Washington University, St. Louis, 1980.

[MON86] Monderer, B. and Lazar, A. A., "A 2-D State Space Model for Speech Communications, I. Inner Ear," *Proceedings of the 20th Annual Conference on Information Sciences and Systems,* Princeton University, Princeton, N. J., pp. 415-420, March 1986.

[MON87] Monderer, B. and Lazar, A. A., "Signal Detection and Resolution in the Cochlea," *Proceedings of the 21th Annual Conference on Information Sciences and Systems,* Johns Hopkins University, Baltimore, MD, pp. 302-307, March 1987.

[NEE81] Neely, S. T., "Finite Difference Solution of a Two-Dimensional Mathematical Model of the Cochlea," *Journal of the Acoustical Society of America,* Vol. 69, No. 5, pp. 1386-1393, 1981.

[RIH69] Rihaczek, A. W., *Principles of High-Resolution Radar,* McGraw-Hill, New York, 1969.

[ROB85] Robinson, D. M., *Transcription of Division 21220 Seminar, 'Hearing-Based Speech Signal Processing' given by Professor David M. Robinson at MRE, with Discussion and Notes,* Transcribed by J. F. Kaiser, Bell Communications Research, Morristown, NJ, April 5, 1985.

[SEN85] Seneff, S., *Pitch and Spectral Analysis of Speech Based on an Auditory Synchrony Model,* Ph. D. Thesis, Massachusetts Institute of Technology, Research Laboratory of Electronics, Technical Report 504, 1985.

[VAN68] Van Trees, H. L., *Detection, Estimation, and Modulation Theory,* Part I, II and III. Wiley and Sons, New York, 1968.

[VET87] Vetterli, M., "A Theory of Multirate Filterbanks," *IEEE Trans. on Acoustics, Speech and Signal Processing,* Vol. ASSP-35, No. 3, March 1987.

[YEG85] Yegnanarayanan, G., *A New Model of Hearing and Its Performance in Pitch Perception,* Ph. D. Thesis, Department of Electrical Engineering, University of Delaware, 1985.

LIMIT CYCLES IN LINEAR SHIFT VARYING DIGITAL FILTERS

TAMAL BOSE AND DAVID P. BROWN
Department of Electrical Engineering
Southern Illinois University
Carbondale, Illinois 62901-6603

ABSTRACT

Magnitude-truncation and 2's complement overflow effects on zero-input shift varying filters are considered. Sufficient conditions on the norm of the system matrix are established such that all zero-input overflow limit cycles are suppressed. Periodically shift-varying filters are considered in detail. Several examples of overflow stable realizations are presented.

I. INTRODUCTION

In digital filter implementation, coefficient quantization and adder overflow introduce nonlinearities in the filter structure. These nonlinear effects can result in large amplitude autonomous oscillations or limit cycles, even though the filter is linearly stable. Limit cycles in linear shift-invariant digital filter structures have been investigated by several researchers [1]-[5]. Different types of arithmetic and structures have been studied in order to suppress these undesirable oscillations. C.W. Barnes and A.T. Fam [6] pointed out that the norm of the system matrix is an important design parameter with respect to overflow behavior for shift-invariant filters. In this paper, a norm is used to investigate limit cycles in shift varying filters.

Some properties useful for studying limit cycles due to magnitude-truncation and adder overflow in shift varying linear digital filters are established. Sufficient conditions are derived for the suppression of zero-input limit cycles (Theorem 1). Similar conditions are established (Theorem 2 and 3) for periodic shift-varying (PSV) digital filters, which find applications in multirate signal processing and other areas where periodically time-varying signals need to be processed [7]-[10]. Overflow stable realizations are presented with examples of second order systems (Section V).

II. STATE SPACE REPRESENTATION

Consider a shift-varying filter realization with the familiar state equations

$$x(n+1) = A(n)x(n) + B(n)U(n) \tag{1}$$

$$y(n) = C(n)x(n) + D(n)U(n) \tag{2}$$

where $n = 0,1,2,\ldots$, $A(n)$, $B(n)$, $C(n)$, and $D(n)$ are of order pxp, px1, 1xp, and 1x1 respectively. In what follows the zero-input behavior of the filter is of interest and therefore, $U(n) = 0$.

A model of the zero-input system with a quantizer in the loop is shown in Fig. 1. The quantizer is such that each state variable is operated on independently of the others and

$$x(n+1) = Q[w(n+1)] = Q[A(n)x(n)] \tag{3}$$

where $Q[x]$ is a bounded nonlinear operator representing magnitude truncation when $-1 \leq x \leq 1$ and and 2's complement overflow operation when x exceeds this range [11]. In detail,

$$|Q[x_k]| \leq |x_k| \tag{4}$$

where x_k is the kth component of x, and

$$||Q[x]|| \leq \mu||x|| \tag{5}$$

where $\mu>0$ and $||.||$ denotes the l_2 norm on a complex n dimensional inner product space, C defined by

$$\langle x,y\rangle = \sum_{i=1}^{n} \bar{x}_i y_i = x^{\dagger} y \tag{6}$$

The dagger, † denotes transposed conjugate.

It is well known [6], that $\mu = 1$ for 2's complement, 1's complement, sign-magnitude, saturation and zeroing arithmetic. Following an analysis similar to that in [6] for shift varying filters, if

$$\mu||A(n)|| < 1, \ n = 0,1,2,\ldots \tag{7}$$

then for the zero-input system,

$$||x(n+1)|| < ||x(n)|| \tag{8}$$

and no limit cycles can exist. For periodically shift varying filters, where the system matrix $A(n)$ is periodic with period $T \geq 2$, and an integer, the condition in (7) becomes

$$\mu||A(n)|| < 1, \ n = 0,1,2,\ldots,T-1 \tag{9}$$

Thus, as in the shift-invariant case, the norm is a design parameter.

III. CONDITIONS FOR THE ABSENCE OF LIMIT CYCLES

In the following theorem, a sufficient condition for no limit cycles to exist in a shift-varying filter realization is given.

Theorem 1: Consider the realization of Fig. 1, if

$$||A(n)|| \leq \delta < 1 \qquad \text{for all } n > N \tag{10}$$

where N is an integer, then overflow limit cycles cannot exist.

Proof: Due to the assumed nature of the quantizer,

$$\begin{aligned} ||x(n+1)|| &\le ||w(n+1)|| \\ &= ||A(n)\, x(n)|| \\ &\le ||A(n)||\ ||x(n)|| \end{aligned} \qquad (11)$$

Thus, since $||A(n)|| \le \delta < 1$ for all $n > N$,

$$||x(n+1)|| < ||x(n)|| \qquad (12)$$

Therefore the norm necessarily decreases and all zero-input oscillations will die out.

The following example is an illustration of Theorem 1 for a real 2x2 $A(n)$.

Example: Consider the symmetric matrix

$$A(n) = \begin{bmatrix} 0 & \sqrt{\dfrac{2}{n+1}} \\ \sqrt{\dfrac{2}{n+1}} & \dfrac{1}{n+1} \end{bmatrix} \qquad (13)$$

If $\lambda_1(n)$ and $\lambda_2(n)$ are the eigenvalues of $A(n)$, then

$$\begin{aligned} ||A(n)|| &= \max\{|\lambda_1(n)|, |\lambda_2(n)|\} \\ &= \max\{\frac{1}{n+1}, \frac{2}{n+1}\} < 1,\ n \ge 2 \end{aligned} \qquad (14)$$

Thus, (14) satisfies the condition of Theorem 1 and the corresponding system is free from limit cycles.

IV. LIMIT CYCLES IN PSV FILTERS

In this section, the suppression of limit cycles in PSV filters is studied. The system of Fig. 1 is considered for period T, where $T \ge 2$ and an integer. Before studying quantization effects, asymptotic stability of the linear filter (filter without the quantizer) must be guaranteed. The following theorem is presented for this purpose.

Theorem 2: Consider the zero-input PSV system

$$x(n+1) = A(n)x(n) \qquad (15)$$

where $x(n+1)$, $x(n)$ are px1 and $A(n)$ is a pxp periodic matrix with period T, $T \ge 2$ and an integer. If

$$\rho(\hat{A}) < 1 \qquad (16)$$

where $\hat{A} = A(T-1)\ A(T-2)\ldots A(0)$ and $\rho(\hat{A})$ is the spectral radius of $\hat{A}$, then the system is asymptotically stable.

Proof: The solution of (15) is

$$x(n+1) = [\prod_{j=0}^{n} A(n-j)]\ x(0) \qquad (17)$$

$$= A(n)\ A(n-1) \ldots A(0)\ x(0) \tag{18}$$

where $A(n)$ is periodic with period T. Let $n = lT + k$, where l and k are positive integers and consider

$$x(n+1) = A(n) \ldots A(lT) \ldots A[(l-1)T] \ldots A(T-1) \ldots A(0)\ x(0) \tag{19}$$

$$= A(n) \ldots A(lT)\ \hat{A}^{l} x(0) \tag{20}$$

where $\hat{A} = A(T-1)\ A(T-2) \ldots A(0)$. Then

$$||x(n+1)|| \leq ||A(n)|| \ldots ||A(lT)||\ \ ||\hat{A}||^{l}\, ||x(0)|| \tag{21}$$

Let $M = \max\limits_{0 \leq j \leq T-1} ||A(j)||$, then

$$||x(n+1)|| \leq M^{k}\ ||\hat{A}||^{l}\, ||x(0)|| \tag{22}$$

Since $\rho(\hat{A}) < 1$, for $\varepsilon > 0$, there exists a norm [12] such that

$$||\hat{A}|| \leq \rho(\hat{A}) + \varepsilon = \delta < 1 \tag{23}$$

Thus, (22) becomes

$$||x(n+1)|| \leq M^{k}\ \delta^{l}\, ||x(0)|| \rightarrow 0 \text{ as } n \rightarrow \infty \tag{24}$$

The following theorem is the main result of this section. It gives a sufficient condition for suppressing all limit cycles in PSV digital filters.

Theorem 3: Consider the system of Fig. 1 with

$A(n) = A(n+T)$, $T \geq 2$ and an integer. If

$$||A(T-1)||\ \ ||A(T-2)|| \ldots ||A(0)|| \leq \delta < 1 \tag{25}$$

then overflow limit cycles cannot exist.

Proof: It is first shown that the system is linearly asymptotically stable. Consider

$$\hat{A} = A(T-1)\ A(T-2) \ldots A(0)$$

and by (25),

$$||\hat{A}|| \leq ||A(T-1)||\ \ ||A(T-2)|| \ldots ||A(0)|| \leq \delta < 1 \tag{26}$$

Therefore $\rho(\hat{A}) < 1$ and from Theorem 2 the linear system is asymptotically stable. Now, the quantized (nonlinear) system is

$$x(n+1) = Q[A(n)x(n)] \tag{27}$$

and therefore

$$x(n) = Q[A(n-1)Q[A(n-2) \ldots Q[A(1)Q[A(0)x(0)]] \ldots] \tag{28}$$

Due to the assumed nature of the quantizer,

$$||x(n)|| \leq ||A(n-1)||\ \ ||A(n-2)|| \ldots ||A(0)||\ \ ||x(0)|| \tag{29}$$

Setting $n-1 = lT + k$, where l and k are positive integers and using (25) and the decomposition procedure used in the proof of Theorem 2, it follows that

$$||x(n)|| \leq ||A(n-1)|| \ldots ||A(lT)||\ \delta^{l}\ ||x(0)|| \rightarrow 0 \text{ as } n \rightarrow \infty \tag{30}$$

and the conclusion follows.

V. OVERFLOW STABLE PSV FILTER REALIZATIONS

In this section some realizations of PSV digital filters such that they are free of zero-input limit cycles are investigated. Special cases of the quantized second-order system:

$$x(n) + a(n)\, x(n-1) + b(n)\, x(n-2) = 0 \tag{31}$$

where $a(n)$ and $b(n)$ are periodic with period T, are studied. Consider the special case when (31) has real eigenvalues $\lambda_1(n)$ and $\lambda_2(n)$ such that $\lambda_1(n) \neq \lambda_2(n)$ for all $0 \leq n \leq T-1$. Then a realization is

$$A(n) = \begin{bmatrix} \lambda_1(n) & 0 \\ 0 & \lambda_2(n) \end{bmatrix} \tag{32}$$

It is easy to check that $A(n)$ is periodic with period T. An example of such a system with $T = 4$ is

$$x(n) - \frac{1}{2}(\sin \frac{n\pi}{2} + \cos \frac{n\pi}{2})\, x(n-1) + \frac{1}{4}(\sin \frac{n\pi}{2}.\cos \frac{n\pi}{2})x(n-2) = 0$$

The eigenvalues of the above are found to be

$$\lambda_1(n) = \frac{1}{2}\,\mathrm{Cos}\frac{n\pi}{2}\ ,\ \lambda_2(n) = \frac{1}{2}\ \mathrm{Sin}\frac{n\pi}{2}$$

If all the eigenvalues of $A(n)$ as in (32) are inside the unit circle for $0 \leq n \leq T-1$, then $||A(n)|| < 1$ for each n since

$$||A(n)|| = \max\{|\lambda_1(n)|,\ |\lambda_2(n)|\} \tag{33}$$

and the system is overflow stable. This is a generalization of the results in [6] to PSV filters. Note that Theorem 3 gives a less restrictive condition on the eigenvalues of $A(n)$ for overflow stability. That is from Theorem 3,

$$\prod_{n=0}^{T-1} \max\{|\lambda_1(n)|,\ |\lambda_2(n)|\} < 1 \tag{34}$$

is a sufficient condition for no overflow limit cycles for the structure of (32). Thus, only the *product* of the maximum eigenvalues have to be inside the unit circle.

Consider the special case when (31) has complex conjugate eigenvalues $\sigma(n) + jw(n)$, $0 \leq n \leq T-1$. A normal form realization is

$$A(n) = \begin{bmatrix} \sigma(n) & w(n) \\ -w(n) & \sigma(n) \end{bmatrix} \tag{35}$$

Again, it is straightforward to check that $A(n)$ is periodic with period T. An example of such a system for $T = 2$ is

$$x(n) + \frac{1}{4}\,\mathrm{Sin}^2[(2n+1)\frac{\pi}{2}]\ x(n-2) = 0$$

The eigenvalues are

$$\lambda(n) = \pm\, j\frac{1}{2}\,\mathrm{Sin}[(2n+1)\frac{\pi}{2}]$$

Since $||A(n)|| = \sigma^2(n) + w^2(n)$, it follows from Theorem 3 that the realization of (35) will be overflow stable if

$$\prod_{n=0}^{T-1} [\sigma^2(n) + w^2(n)] < 1 \tag{36}$$

For a first-order system,

$$||A(n)|| = |\lambda(n)| \tag{37}$$

and by Theorem 3 a sufficient condition for overflow stability is

$$\prod_{n=0}^{T-1} |\lambda(n)| < 1 \tag{38}$$

An example of such a system for T = 4 is

$$x(n) - (\operatorname{Cos}\frac{n\pi}{2})\, x(n-1) = 0$$

It follows that $\lambda(n) = \operatorname{Cos}\frac{n\pi}{2}$ and

$$||A(0)||\ ||A(1)||\ ||A(2)||\ ||A(3)|| = 0 \tag{39}$$

Therefore, from (38), the system is overflow stable. However, $|\lambda(0)| = |\lambda(2)| = 1$, $|\lambda(1)| = |\lambda(3)| = 0$.

A second-order system with a double pole for each $0 \leq n \leq T-1$ is realized as

$$A(n) = \begin{bmatrix} \lambda(n) & 1 \\ 0 & \lambda(n) \end{bmatrix} \tag{40}$$

and implemented as a cascade of two first-order systems [6]. If (38) is satisfied then the overall system is free from limit cycles.

REFERENCES

[1] P.M. Ebert, J.E. Mazo, and M.G. Taylor, "Overflow oscillations in digital filters," Bell Syst. Tech. J., Vol. 48, pp. 2999-3020, Nov. 1969.

[2] L.B. Jackson, "On the interaction of roundoff noise and dynamic range in digital filters," Bell Syst. Tech. J., Vol. 49, pp. 159-184, Feb. 1970.

[3] Theo A.C.M. Classen and Lars O.G. Kristiansson, "Necessary and sufficient conditions for the absence of overflow phenomena in a second-order recursive digital filter," IEEE Trans. Acous. Speech & Signal Process., Vol. ASSP-23, No. 6, pp. 509-519, Dec. 1975.

[4] W.L. Mills, C.T. Mullis,, and R.A. Roberts, "Low roundoff noise and normal realizations of fixed point IIR digital filters," IEEE Trans. Acous. Speech & Signal Process., Vol. ASSP-29, No. 4, pp. 893-903, Aug. 1981.

[5] C.L. Phillips and H.T. Nagle, "Digital control system analysis and design," Prentice-Hall, Inc., NJ, 1984.

[6] C.W. Barnes and A.T. Fam, "Minimum norm recursive digital filters that are free of overflow limit cycles," IEEE Trans. Circuits & Syst., Vol. CAS-24, No. 10, pp. 569-574, Oct. 1977.

[7] R.A. Meyer and C.S. Burrus, "A unified analysis of multirate and periodically time varying digital filters," IEEE Trans. Circuits & Syst., Vol. CAS-22, No. 3, pp. 162-168, Mar. 1975.

[8] N. Huang and J.K. Aggarwal, "On linear shift-variant digital filters," IEEE Trans. Circuits & Syst., Vol. CAS-27, No. 8, pp. 672-678, Aug. 1980.

[9] R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc. N.J., 1983.

[10] T. Bose and D.P. Brown, "On the stability of linear shift-variant digital filters," Proc. Intl. Conf. Acous. Speech & Signal Process., Vol. 2, pp. 880-883, April 1987.

[11] W.L. Mills, C.T. Mullis, and R.A. Roberts, "Digital filter realizations without overflow oscillations," IEEE Trans. Acous. Speech and Signal Process., Vol. ASSP-26, No. 4, pp. 334-338, Aug. 1978.

[12] J.M. Ortega and W.C. Rheinholt, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, p. 44, 1970.

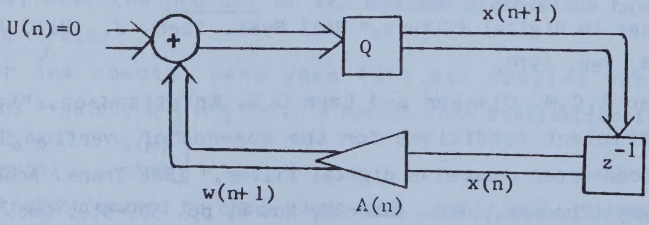


Figure 1. State Space Model with Quantizer

Best Orderings for Floating Point Addition

T.G. Robertazzi and S.C. Schwartz,
Depts. of Electrical Engineering,
SUNY at Stony Brook and Princeton University

I. Introduction

In his pioneering work on finite precision errors [1], Wilkinson pointed out that the error bounds for floating point addition are smallest, among all orderings, if the numbers are added in order of increasing magnitude. The following will demonstrate, through a simple stochastic analysis, that if the storage of various intermediate sums is allowed, as well as simple ordering, then even more accurate results can be obtained.

II: A Stochastic Analysis

2.1 Simple Ordering

In the stochastic analysis of this work, the relative errors, ϵ, made in performing the addition will be assumed to be statistically independent, zero mean and to possess a variance (mean square error) of σ_ϵ^2 . Various ordering schemes are compared on the basis of the mean square error, σ_E^2, of the final sum.

The analysis is made using the uniform and exponential distributions to describe the numbers being added. Naturally this ensures that the analysis is tractable. Numbers involved in actual calculations may or may not be adequately described in terms of independent random variables from these particular distributions. If these distributions are parametrized as U(0,1) and Expo ($\mu = \sigma = 0.5$) , then both distributions have the same mean and variances of $\frac{1}{12}$ and $\frac{1}{4}$, respectively.

In this analysis, the M numbers generated by these distributions will be approximated by the means of their order statistics. For the uniform distribution the expected value of the i $^{\text{th}}$ order statistic is:

$$\frac{i}{M+1} \tag{2.1}$$

For the exponential distribution, with mean μ, it is:

$$\mu \sum_{j=1}^{i} \frac{1}{M+1-i} \tag{2.2}$$

For this simple stochastic analysis, the noise corrupted intermediate sums will be replaced by the expected values of these noisy quantities. Since the errors are zero mean, this is suitable for a first order approximation. It is then a matter of some algebra to calculate the mean square error of the final sum. Tabulating the results for increasing, decreasing and random orderings [2], one has:

Table 1

This research was supported in part by the O.N.R. under contract N00014-81-K-0146.

Simple Ordering			
Distribution	Increasing	Random	Decreasing
U(0,2 μ)	$.2\mu^2M^3\sigma_\varepsilon^2$	$.33\mu^2M^3\sigma_\varepsilon^2$	$.53\mu^2M^3\sigma_\varepsilon^2$
Expo (μ)	$.13\mu^2M^3\sigma_\varepsilon^2$	$.33\mu^2M^3\sigma_\varepsilon^2$	$.63\mu^2M^3\sigma_\varepsilon^2$

The ranking of performance in this table is as one might expect. Adding the numbers in accordance with increasing magnitude is the best policy. The improvement in doing this, as opposed to ordering in accordance with decreasing magnitude, is greater for the exponential distribution.

2.2: Ordering with Intermediate Storage

In order to minimize inaccuracies due to errors in floating point addition, numbers of similar magnitude should always be added. One very natural way of achieving this is as follows. The numbers are first ordered in accordance with increasing magnitude. The two smallest are added and this intermediate sum is inserted among the remaining, ordered, numbers so that the monotonicity is maintained. This process is repeated until the final sum is produced in this *insertion adder.*.

To develop an expression for the mean square error in the final sum, assume that at the i^{th} iteration the exact sum has been evenly divided among the remaining M+1-i numbers. At the i^{th} iteration one is adding two numbers with magnitudes of:

$$\frac{M\mu}{M-i+1} \tag{2.3}$$

The mean square error in the final sum is then:

$$\sigma_E^2 \approx \sum_{i=2}^{\infty}\left(\frac{M\mu}{i}+\frac{M\mu}{i}\right)^2\sigma_\varepsilon^2 \tag{2.4}$$

$$\sigma_E^2 \approx 4\mu^2M^2\sigma_\varepsilon^2\sum_{i=2}^{\infty}\frac{1}{i^2} \approx 2.6\mu^2M^2\sigma_\varepsilon^2 \tag{2.5}$$

Note that the mean square error now grows as a quadratic function of M, rather than the cubic dependence exhibited by simple ordering. Because it is assumed that the sum is always *equally* divided among the remaining numbers, the above growth rate is actually an upper bound.

In an *adjacency adder* the numbers are ordered and adjacent numbers are continually added in parallel. Based on a similar analysis [2] one has:

Table 2

Ordering + Intermediate Storage		
	Insertion	Adjacency
$U(0,2\mu)$	2.6 $\mu^2M^2\sigma_\varepsilon^2$	2.7 $\mu^2M^2\sigma_\varepsilon^2$
Expo (μ)	2.6 $\mu^2M^2\sigma_\varepsilon^2$	4 $\mu^2M^2\sigma_\varepsilon^2$

III: Stochastic Simulations

To verify the validity of the expressions for the mean square error of the final sum which were developed in the previous section, several Monte Carlo simulations were performed. There was general agreement between these expressions and the simulations [2].

IV. Conclusion

The performance of the insertion and adjacency adder demonstrate that considering only simple processing order is overly restrictive in the context of the problem of minimizing the mean square error in finite precision, floating point addition.

References

[1] J.H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, 1963, pg. 17.

[2] T.G. Robertazzi, *Measurement Processing Order and Recursive Statistical Estimation*, Ph.D Thesis, Princeton University, 1981. Extended manuscript available upon request.

A RATIONAL VIEW OF THE FAST FOURIER TRANSFORM

CHARLES M. FIDUCCIA
General Electric R&D Center
Schenectady, New York 12301

Nearly two decades ago, the fast Fourier transform (FFT) [1] revolutionized the computation of the discrete Fourier transform (DFT). Most explanations of the FFT are in terms of matrix factorization. In 1972, the author [2] showed how the FFT could be viewed as repeated polynomial division. Also see [3] for an account of this method. The present paper shows that the DFT has a beautiful representation as a very simple rational function, and that performing the DFT consists of transforming this rational function from partial-fraction form into normal form. In this setting, the FFT is seen as nothing more than a rational addition tree in which special attention is paid to the order in which the partial fractions are assigned to the leaves of the tree for the purpose of maintaining very sparse denominators in all intermediate partial sums. This new approach gives high-level algebraic insight into the computation that can serve as the basis of new FFT algorithms.

The DFT of a (column) vector $a = [a_0, \ldots, a_{n-1}]^T$ is the vector $f = [f_0, \cdots, f_{n-1}]^T$, where $f_i = \sum_{j=0}^{n-1} a_j \omega^{ij}$ and ω is a primitive n^{th} root of unity. Thus $f = \Omega a$, where $\Omega = [\omega^{ij}]$ is an $n \times n$ matrix indexed by $i, j = 0, \ldots, n-1$. To make the vectors f and a more algebraically tractable, we convert them into polynomials by defining $a(x) = \sum_{j=0}^{n-1} a_j x^j$ and $f(x) = \sum_{i=0}^{n-1} f_i x^{n-1-i}$. The division method in [2] was based on the observation that $f_i = a(\omega^i)$; here, we concentrate on the output polynomial $f(x)$. Substituting for f_i in $f(x)$, we have

$$f(x) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j \, \omega^{ij} \, x^{n-1-i} = \sum_{j=0}^{n-1} a_j \, x^{n-1} \sum_{i=0}^{n-1} (\omega^j/x)^i = \sum_{j=0}^{n-1} a_j \, L_j(x)$$

$$L_j(x) = x^{n-1} \cdot \frac{(\omega^j/x)^n - 1}{\omega^j/x - 1} = x^{n-1} \cdot \frac{1 - x^n}{x^n} \cdot \frac{x}{\omega^j - x} = \frac{x^n - 1}{x - \omega^j}$$

$$f(x) = (x^n - 1)\phi(x) \quad \text{where} \quad \phi(x) = \sum_{j=0}^{n-1} \frac{a_j}{x - \omega^j}$$

Since ω is a primitive n^{th} root of unity, the n constants ω^j in the denominators of $\phi(x)$ are all of the n roots of unity. The product of the n denominators is thus $\prod_{j=0}^{n-1} (x - \omega^j) = x^n - 1$, proving that the normal form of $\phi(x)$ is $f(x)/(x^n - 1)$. This establishes the following result.

Theorem 1 *The DFT of* $(a_0, \ldots, a_{n-1})$ *is given by the coefficient sequence* $(f_0, \ldots, f_{n-1})$ *of the numerator* $f(x) = \sum_{i=0}^{n-1} f_i \, x^{n-1-i}$ *in the normal form* $f(x)/(x^n - 1)$ *of the expression*

$$\phi(x) = \sum_{j=0}^{n-1} \frac{a_j}{x - \omega^j}$$

Although our derivation was ad hoc, $\phi(x)$ can also be obtained, using the Lagrange interpolation formula and the inverse DFT, from the observation that $f(\omega^j) = na_j/\omega^j$. We also note that $\phi(z) = F(z)z^{n-1}/(z^n - 1)$, where $F(z) = \sum_{i=0}^{n-1} f_i \, z^{-i}$ is the Z transform of $f_0, \ldots, f_{n-1}$.

Using Theorem 1, we propose to compute the DFT by simply adding the n partial fractions in $\phi(x)$ and looking at the resulting numerator. Rational addition will be done using the familiar rule $\frac{p(x)}{r(x)} + \frac{q(x)}{s(x)} = \frac{p(x)s(x)+q(x)r(x)}{r(x)s(x)}$ under the assumption that scalar arithmetic operations have unit cost. The only freedom we have, in this "just-add-them" class of algorithms, is the order in which partial sums are combined. This is equivalent to specifying a binary addition tree with n

leaves and assigning the n initial partial fractions to the leaves. The rational expression at each node then represents an intermediate partial sum, and the root represents $\phi(x)$.

Although all such trees (algorithms) compute $\phi(x)$, different trees may have different costs. Our goal is to show that this class contains "fast" $O(n \log n)$ algorithms. For simplicity, let us confine our attention to *the* binary tree of height h on $n = 2^h$ leaves. The only freedom left with this tree is to decide the order in which the partial fractions should be assigned to the leaves. Even within this restricted class, costs range from $O(n \log n)$ to $O(n^2)$ because the cost of a tree is determined by how sparse the denominators are at each of its internal nodes. If we define the *length* $L(p)$ of a polynomial $p(x)$ to be the number of monomials in $p(x)$, we have

Lemma 1 *Let $L(p) = l$, and $L(q) = m$. If both $L(r)$ and $L(s)$ are bounded by some constant, then $\frac{p(x)}{r(x)} + \frac{q(x)}{s(x)} = \frac{p(x)s(x)+q(x)r(x)}{r(x)s(x)}$ can be computed with with $O(l + m)$ arithmetic operations.*

Lemma 2 *Let v be any node of any addition tree T for $\phi(x)$. If the subtree S rooted at v has m leaves, then the numerator at v is of degree at most m and the denominator at v is the product of the m denominators at the leaves of S.*

If we concentrate just on the denominators at all the nodes of T, Lemma 2 says that T is a *factor tree* for $x^n - 1$. The problem of finding good leaf assignments can thus be attacked in a top-down fashion [2]. Simply start with $x^n - 1$ at the root, factor it into two sparse factors and use them as the denominators of the sons of that node. Do this recursively down to the leaves.

One possible *factoring scheme* is based on the "stable-under-factoring" form $x^{2m} \pm c$. Note that $x^{2m} - c^2 = (x^m - c)(x^m + c)$, and that the root $x^n - 1$ is of this form. When $n = 2^h$, this gives an addition tree of height h in which every denominator has length 2. The rational addition at every node v (which is the root of some subtree S with $m \geq 2$ leaves) is of the form

$$\frac{p(x)}{x^m - c} + \frac{q(x)}{x^m + c} = \frac{(p(x) + q(x))x^m + (p(x) - q(x))c}{x^{2m} - c^2}$$

This corresponds to computing a set of m "butterflies" in the FFT. Applying the above lemmas to each node of this tree shows that the total cost is $O(n \log n)$. Since the two factors at any internal node of T can be rotated, we get 2^{n-1} variations of the tree, all doing the same operations. One of the trees obtained in this way has the partial fractions of $\phi(x)$ assigned to its leaves in bit-reversed order. Thus any rotation of this order also has cost $O(n \log n)$.

Because the computational efficiency of this method is based purely on the idea of sparse factorings, one may look for other stable denominator forms. An example, of length 3, is $(x^{2m} + b) \pm ax^m$, since $((x^{2m} + b) + ax^m)((x^{2m} + b) - ax^m) = (x^{4m} + b^2) + (2b - a^2)x^{2m}$. This yields another binary addition tree for the DFT whose cost is also $O(n \log n)$. Indeed, we have

Theorem 2 *Let T be any factor tree for $x^n - 1$ that has factors of bounded lengths at all of its nodes. If T has height h, then T represents an $O(h\,n)$ DFT algorithm.*

This applies to the above rational-sum method as well as the division method [2]. A more general version expresses the cost of T in terms of its external path length. Within this class, many trees of various costs can be constructed by combining factoring schemes. By using $\phi(x)$ in other ways, it may be possible to derive totally new – possibly $O(n)$ – FFT algorithms.

References

[1] Cooley, J. M. and J. W. Tookey, "An algorithm for the machine computation of complex Fourier series," *Math. Comp.* vol. 19, April 1965, pp. 297-301.

[2] Fiduccia, C. M., "Polynomial evaluation via the division algorithm: the fast Fourier transform revisited," *Proc. 4th Annual ACM Symp. Theory of Computing*, May 1972, pp. 88-93.

[3] Aho, A.V., J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading Massachusetts, 1974.

SPLIT LEVINSON ALGORITHM FOR TOEPLITZ MATRICES OF ANY RANK PROFILE

M.KEMAL CILIZ and HARI KRISHNA
Electrical and Computer Engineering Department
Syracuse University, Syracuse, NY 13244-1240

ABSTRACT

In this work, the Split Levinson algorithm is applied to linear systems with real-symmetric Toeplitz matrices of *any rank profile*, namely systems with singular sub-matrices.

1. SPLIT LEVINSON ALGORITHM FOR ANY RANK PROFILE

The solution for a linear system of equations where the coefficient matrix is Toeplitz, is given by the well known Levinson Algorithm(LA) [1]. A more general algorithm for the solution of Hermitian Toeplitz matrices with any rank profile is introduced in [2]. Recently a modified form of LA, referred as Split Levinson Algorithm(SLA), is proposed in [3] and shown to be twice more economical than the classical LA regarding the number of multiplications. The algorithm presented in this paper is the combination of these two algorithms for the case of a real-symmetric Toeplitz matrix. Only the symmetric version of the SLA is introduced in this text, whereas the anti-symmetric version of the algorithm is similar in nature and given in [4].

Given the real-symmetric Toeplitz matrix C_m with its submatrices C_k defined as,

$$C_k=[c_{i-j} : 0 \leq i,j \leq k-1] \text{ with } c_{-i}=c_i \text{ ,for } k=1,2,\ldots,m. \tag{1}$$

and the determinants denoted by $f_k=\det(C_k)$ with the convention $f_0=1$. The linear system of equations at the k th step are given as ,

$$C_k a_{k-1}=(1,0,\ldots,0)^T \qquad \text{with } a_{k-1}=f_{k-1}/f_k \tag{2}$$

$$C_{k+1} x_k=(\sigma_{k+1},0,0,\ldots,0)^T \quad \text{with } x_{k,0}=1 \text{ and } \sigma_{k+1}=f_{k+1}/f_k \tag{3}$$

Note that $x_k(z)=\sigma_{k+1}a_k(z)$ assuming $\sigma_{k+1}\neq 0$. The reflection coefficient e_k is given by,

$$e_k= -\Sigma_{i=0}^{k-1} (c_{k-i}a_{k-1,i}) \tag{4}$$

The symmetric version of SLA implies the following equations,

$$\tau_k = \Sigma_{i=0}^{t-1} (c_i+c_{k-i})p_{k,i} \qquad \text{if } k=2t-1 \tag{5}$$

or

$$\tau_k = \Sigma_{i=0}^{t-1} (c_i+c_{k-i})p_{k,i} + c_t p_{k,t} \quad \text{if } k=2t \tag{6}$$

$$p_{k+1,i}=p_{k,i}+p_{k,i-1}-\alpha_k p_{k-1,i-1} \quad \text{for } 1 \leq i \leq t \tag{7}$$

$$\alpha_k=\tau_k/\tau_{k-1}, \ \ \tau_k=\sigma_{k+1}/\lambda_k \text{ and } \lambda_k=2-\alpha_k/\lambda_{k-1} . \tag{8}$$

If $x_k(z)$ is needed at this step, it's given as

$$x_{k,i}=x_{k,i-1}+p_{k+1,i}-\lambda_k p_{k,i-1} \qquad \text{for } i=1,2, \ldots k, \text{ with } x_{k,0}=1 \tag{9}$$

Given the initials conditions for $p_{0,0}$, $p_{1,0}$, $p_{1,1}$, τ_0 and λ_0, one can start evaluating the singular predictor polynomials $p_k(z)$ and then $x_k(z)$. But the SLA fails whenever one encounters a singular sub-matrix C_k. When $\tau_k=0(\lambda_k=2)$ or $\lambda_k=0(\tau_k\neq 0)$, then $\sigma_{k+1}=\tau_k\lambda_k=0$ which implies that C_{k+1} is singular. This means there are two cases to be determined at a singular point.

Case1: $\tau_k=0$ $(\lambda_k=2)$--> $\sigma_{k+1}=\tau_k\lambda_k=0$. Then, $e_k=\lambda_k-1=1$. From [3] it is known that at a singular point $p_k(z)=x_{k-1}(z)+zx_{k-1}(z)$. THIS IMPLIES $x_k(z)=p_k(z)$. Using the algorithm given in [2], next non-singular point is found as follows: First find the smallest integer l satisfying,

$$c_l x_{k,0}+c_{l+1}x_{k,1}+ \cdots +c_{l+k}x_{k,k}\neq 0=\beta_0 \tag{10}$$

Once l is found, the next non-singular point is given by, $n=k+2l$. Next, one should find $x_n(z)$ and $a_{n-1}(z)$ in order to continue with the SLA.

$$a_{n-1}(z)=(\beta_0)^{-1}z^l x_k(z) , \quad \text{with } \beta_0 \text{ as found in (10)} \tag{11}$$

$$x_n(z)=\beta_0(z^l a_{k-1}(z)+p'(z)x_k(z)) , \qquad \text{where } a_{k-1}(z)=x_{k-1}(z)/\sigma_k \tag{12}$$

$p'(z)=p'_0+p'_1z+ \ldots +p'_{2l}z^{2l}$. The coefficients are obtained through the following equations,

$$\begin{bmatrix} \Omega_0 & & & \\ \Omega_1 & \Omega_0 & & 0 \\ & \ddots & \ddots & \\ \Omega_l & & \Omega_1 & \Omega_0 \end{bmatrix} \begin{bmatrix} p'_0 \\ p'_1 \\ \vdots \\ p'_l \end{bmatrix} = - \begin{bmatrix} \mu_l \\ \mu_{l-1} \\ \vdots \\ \mu_1 \end{bmatrix} \tag{13}$$

$$\mu_i = \sum_{j=0}^{k-1} c_{k+l-i-j} a_{k-1,j} \qquad 0 \leq i \leq l, \tag{14}$$

$$\Omega_i = \sum_{j=0}^{k} c_{k+l+i-j} x_{k,j} \qquad 0 \leq i \leq l. \tag{15}$$

The last l coefficients of p'(z) are obtained through the relation $p'_{l+i} = -p'_{l-i}$ for i=1,2, ... l. Now we check σ_{n+1} to see if the next point n+1 is a singular point or not, σ_{n+1} is given as

$$\sigma_{n+1} = \beta_0^2 (2p'_l + a_{k-1}(0)) \quad \text{where } a_{k-1}(z) = x_{k-1}(z)/\sigma_k \tag{16}$$

If $\sigma_{n+1}=0$, one should go back to the beginning of this algorithm (Case 1), if $\sigma_{n+1} \neq 0$ then one continues with the SLA. In order to continue with SLA, $p_n(z)$ and $p_{n+1}(z)$ are found by the following equations,

$$p_n(z) = (x_n(z) + \hat{x}_n(z))/\lambda_k \tag{17}$$

$$p_{n+1}(z) = x_n(z) + z\hat{x}_n(z) \tag{18}$$

where $\hat{x}_n(z) = z^n x_n(z^{-1})$.

Case 2: $\lambda_k = 0 (\tau_k \neq 0)$ ---> $\sigma_{k+1} = \tau_k \lambda_k = 0$ ---> $e_k = \lambda_k - 1 = -1$. This implies we are again at a singular point. First one obtains $x_k(z)$ using $p_k(z)$ and $p_{k+1}(z)$,

$$x_k(z) = (p_{k+1}(z) - \lambda_k z p_k(z))/(1-z) \;\text{--->}\; x_k(z) = p_{k+1}(z)/(1-z) \quad \text{since } \lambda_k = 0 \tag{19}$$

Now, one should find the next non-singular point using (10). At the non-singular point n, $x_n(z)$ is obtained through (11)-(15). The rest of the algorithm is the same as in Case 1.

Example: We work on the same example given in [2]. Given C_8 (m=8) with $c_0=c_1=0$, $c_2=c_3=c_4=1$, $c_5=2$, $c_6=c_7=3$. The initial values are given as $p_{00}=2$, $p_{11}=1$, $\tau_0=c_0$, $\lambda_0=1$, $a_{-1}=0$, $\sigma_0=c_0$. Since $\sigma_0=0$, (10) is used to obtain $l=2$ and $\beta_0=1$. The next non-singular point is $n=k+2l=4$. Using (13)-(15), p'(z) is obtained as $p'(z)=1-z+z^3-z^4$. Then using (11),(12) and (16) with n=4, one obtains $a_3(z)=z^2$, $x_4(z)=1-z+z^3-z^4$ and $\sigma_5=0$. Since $\sigma_5=0$, this means next point is also singular. Then one jumps to the next non-singular point $n=k+2l=6$, using $l=1$ and $\beta_0=-2$ by (10). Now $p'(z)=(-2-z+2z^2)/4$ and $a_5(z)$, $x_6(z)$, σ_7 are given by (11),(12) and (16) as, $a_5(z)=-(z-z^2+z^4-z^5)/2$, $x_6(z)=(2-z-3z^2-z^4-3z^5+2z^6)/2$ and $\sigma_7=-2$. Since $\sigma_7 \neq 0$, we go back to the SLA. Then by (17), $p_6(z)=(1-z-z^2-z^4-z^5+z^6)$ and by (18), $p_7(z)=(2+z-6z^2-z^3-z^4-6z^5+z^6+2z^7)/2$. We need $p_8(z)$ to calculate $x_7(z)$. Using (5),(7) and (8), $p_8(z)=1-4z+3z^2+2z^3-z^4+2z^5+3z^6-4z^7+z^8$. Then next step is to calculate the final result $x_7(z)$ by (9). $x_7(z)=(8-18z+9z^2+7z^3-4z^4+9z^5+15z^6-14z^7)/8$.

2. CONCLUSIONS

The algorithm presented above briefly discusses the symmetric version of SLA for real-symmetric Toeplitz matrices of any rank profile. The algorithm is more efficient than the classical LA regarding the number of multiplications and is applicable to systems with singular sub-matrices.

References:

[1] N.Levinson, "The Wiener rms (root mean square) error criterion in filter design and prediction,"*J.Math. Phys.*, vol. 25, pp. 261-278, 1946.

[2] P.Delsarte, Y.V. Genin, Y.G. Kamp, "A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, no.4, pp. 964-971, 1985.

[3] P.Delsarte, Y.V.Genin, "The Split Levinson algorithm," *IEEE Trans. ,Acoust., Speech, Signal Processing* , vol ASSP-34, no.3, 1986.

[4] K.Ciliz and H.Krishna, "Generalized Split Levinson algorithm", (*in preparation*).

AN ALGORITHM FOR ESTIMATING THE FREQUENCIES AND AMPLITUDE MATRICES CORRESPONDING TO MULTIDIMENSIONAL HARMONIC SIGNALS IN NOISE

C. FOIAS
Department of Mathematics, Indiana University, Bloomington, IN 47405
A. E. FRAZHO
School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907
P. J. SHERMAN
School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907

ABSTRACT In this paper we present a new recursive algorithm for estimating both the frequencies and amplitude matrices corresponding to a p-dimensional harmonic signal in additive noise of arbitrary color.

1. INTRODUCTION

In this paper we present a new recursive algorithm for estimating both the frequencies and the amplitude matrices corresponding to a p-dimensional harmonic signal. The results we obtain provide a generalization of the results in [FFS] for the one-dimensional setting. As such, they are valid for any regular colored noise environment.

We begin in Section 2 by developing the Naimark dilation, U for the correlation sequence corresponding to the p-dimensional observation process. The Wold decomposition of U and the multidimensional Levinson polynomials are used in Section 3 to completely characterize the eigenvalues and corresponding eigenvectors of U*. The recursive algorithm for estimating the signal frequencies and amplitude matrices follows immediately from this characterization. It is described in detail in Section 4.

2. PRELIMINARIES

We begin by introducing some notation which will be used throughout this paper. All spaces are Hilbert spaces. The Hardy space $H^2(\mathscr{H})$ corresponding to any Hilbert space $\mathscr{H}$ is the subspace of $L^2([0,2\pi],\mathscr{H})$ whose Fourier coefficients of $e^{in\omega}$ are zero for $n < 0$; that is, h is an element of $H^2(\mathscr{H})$ if

$$h(e^{i\omega}) = \sum_0^\infty h_n e^{in\omega} \quad \text{and} \quad \|h\|^2_{H^2} = \|h\|^2_{L^2} = (2\pi)^{-1}\int_0^{2\pi} \|h(e^{i\omega})\|^2_{\mathscr{H}}\, d\omega = \sum_0^\infty \|h_n\|^2_{\mathscr{H}} < \infty\ .$$

Henceforth, we will use subscripts on norms and inner products only when confusion might otherwise occur. The open unit disk will be denoted by $D = \{z \epsilon C : |z| < 1\}$.

Consider the p-dimensional observation process

$$X_n = S_n + N_n$$

where N_n is a purely nondeterministic wide sense stationary zero mean random noise process, and S_n, uncorrelated with N_n, is a purely deterministic random signal process of the form

$$S_n = \sum_{m=1}^{k} A_m\, [\cos(n\omega_m + \xi_{1m}),\ldots,\cos(n\omega_m + \xi_{pm})]^t. \tag{2.1}$$

Here the $\{A_m\}_1^k$ are nonzero p x p matrices with real elements and the $\{\xi_{jm} : 1 \leq j \leq p \text{ and } 1 \leq m \leq k\}$ are mutually independent random variables uniformly distributed over $[0,2\pi]$. It then follows that S_n is also a wide sense stationary zero mean random process. The purpose of this paper is to develop a recursive algorithm for estimating the set of *distinct* signal frequencies $\Omega = \{\omega_m\}_1^k$ and the corresponding amplitude matrices $\{A_m\}_1^k$. In the remainder of this section we develop the framework in which this problem will be addressed. This development will generalize that of the one-dimensional setting in [FFS].

The correlation function for S_n is given by

$$R_S(n) = E(S_j S_{j+n}^t) = \frac{1}{2}\sum_{1}^{k} A_m A_m^* \cos(n\omega_m) \tag{2.2}$$

where * denotes the adjoint. Since we only depend on the wide sense stationarity of S_n, without loss of generality we assume that A_m is positive (≥ 0) hence that $A_m A_m^* = A_m^2$. Using the DeMoivre-Euler identity it is easy to show that

$$R_S(n) = C_u W^n C_u^* \qquad \text{(for all n)} \tag{2.3}$$

where

$$C_u = \frac{1}{2}\ [A_1, A_1, A_2, A_2, \ldots, A_k, A_k]$$

and

$$W = \text{diagonal}\ [e^{i\omega_1} I_1, e^{-i\omega_1} I_1, e^{i\omega_2} I_2, e^{-i\omega_2} I_2, \ldots, e^{i\omega_k} I_k, e^{-i\omega_k} I_k]$$

is on

$$\bigoplus_{1}^{k} [R(A_m) \oplus R(A_m)]\ .$$

Here I_m is the identity on the range space of A_m, denoted $R(A_m)$. Finally, notice that W has 2k distinct eigenvalues appearing in the following order: $e^{i\omega_1}, e^{-i\omega_1}, \ldots, e^{i\omega_k}, e^{-i\omega_k}$.

Since the noise process N_n is purely nondeterministic its correlation function has a spectral representation of the form

$$R_N(n) = (2\pi)^{-1} \int_0^{2\pi} S_N(e^{i\omega})\, e^{in\omega} d\omega. \tag{2.4}$$

Throughout this paper we will assume that $S_N(e^{i\omega})$ admits a full rank outer (or minimum phase) spectral factor $\Theta(z)$; that is $S_n(e^{i\omega}) = \Theta(e^{i\omega})^*\ \Theta(e^{i\omega})$ where $\Theta(z)$ is a p x p matrix-valued analytic function in D satisfying

$$\overline{\Theta H^2(\mathscr{H})} = H^2(\mathscr{H})\ .$$

Here the bar denotes the closure, and $\mathscr{H}$ denotes the usual p-dimensional complex vector space C^p utilizing the standard inner product. We remark that many of the results in this paper can easily be extended to an arbitrary Hilbert space $\mathscr{H}$.

To obtain a representation for $R_N(n)$ analogous to that of $R_S(n)$ given by (2.3) let S denote the unilateral shift on $H^2(\mathscr{H})$; that is, $Sh = e^{i\omega}h$ for all h in $H^2(\mathscr{H})$. We then have

$$R_N(n) = C_0 S^n C_0^* \qquad \text{(for all } n \geq 0) \tag{2.5}$$

where the operator C_0^* mapping $\mathscr{H}$ into $H^2(\mathscr{H})$ is defined by

$$C_0^* h = \Theta(e^{i\omega})h \qquad (h \epsilon \mathscr{H})\ . \tag{2.6}$$

To compute C_0 and verify (2.5) consider any h in $\mathscr{H}$ and g in $H^2(\mathscr{H})$. Then

$$(h, C_0 g)_{\mathscr{H}} = (C_0^* h, g)_{H^2} = (\Theta h, g)_{H^2} = (h, \Theta^* g)_{L^2} =$$

$$(2\pi)^{-1} \int_0^{2\pi} (h, \Theta^* g)_{\mathscr{H}}\, d\omega = (h, (2\pi)^{-1} \int_0^{2\pi} \Theta^* g\, d\omega)_{\mathscr{H}}\ .$$

Therefore the operator C_0 mapping $H^2(\mathscr{H})$ into $\mathscr{H}$ is given by

$$C_0 g = (2\pi)^{-1} \int_0^{2\pi} \Theta\ (e^{i\omega})^* g(e^{i\omega}) d\omega \qquad (g \epsilon H^2(\mathscr{H}))\ . \tag{2.7}$$

Clearly (2.6) and (2.7) along with the definition of Θ yield the representation (2.5) for $R_N(n)$.

The signal and noise processes are assumed to be independent of one another so that using (2.3) and (2.5) the correlation function for the observation process X_n has the representation

$$R_X(n) = R_S(n) + R_N(n) = CU^nC^* \qquad (\text{for } n \geq 0) \tag{2.8}$$

where

$$C = [C_u\ ,\ C_0]$$

and the operator

$$U \triangleq W \oplus S \quad \text{on } \mathscr{K} \triangleq \Big(\bigoplus_1^k [R(A_m) \oplus R(A_m)]\Big) \oplus H^2(\mathscr{H})$$

is an isometry. Notice that C^* maps $\mathscr{H}$ into $\mathscr{K}$, and that $\mathscr{K}$ is the closed linear span of $\{U^nC^*\mathscr{H}\}_0^\infty$, that is,

$$\mathscr{K} = \bigvee_0^\infty U^nC^*\mathscr{H}\ . \tag{2.9}$$

The isometry U is called the *Naimark dilation* for $\{R_X(n)\}$, and the decomposition of U into $W \oplus S$ is called the *Wold decomposition* of U. (See Chapter I of [SF] for further details.)

As in the one-dimensional setting, the problem of identifying the set of signal frequencies Ω is equivalent to identifying all the eigenvalues of U. This follows because the spectrum of the unilateral shift S has no eigenvalues [H1]. So the eigenvalues of U are precisely the eigenvalues of W. In the following section we will use the Wold decomposition and the multidimensional Levinson algorithm [MVK], to completely characterize the eigenvalues and corresponding eigenvectors of U^*, and in particular W^*.

3. THE EIGENVALUES AND EIGENVECTORS OF U*

In this section we give a complete characterization of the eigenvalues and eigenvectors of U^*. This will produce an algorithm for computing estimates of the signal frequencies Ω and the corresponding amplitude matrices $\{A_m\}$.

To begin, define the nested sequence of subspaces

$$\mathscr{K}_n = \bigvee_0^n U^jC^*\mathscr{H}\ . \tag{3.1}$$

Clearly $\mathscr{K}_n$ is strictly contained in $\mathscr{K}_{n+1}$, $(\mathscr{K}_n \subset \mathscr{K}_{n+1})$, for if $\mathscr{K}_n = \mathscr{K}_{n+1}$ this would contradict the infinite dimensionality of $H^2(\mathscr{H}) \subset \mathscr{K} = \mathscr{K}_\infty$. Let $P_{\mathscr{K}_n}$ denote the orthogonal projection onto $\mathscr{K}_n$. Then we can write

$$\Psi_n(U)h \triangleq U^nC^*h - P_{\mathscr{K}_{n-1}}U^nC^*h = \sum_0^n U^{n-j}C^*B_n(j)h \qquad (h \epsilon \mathscr{H}) \tag{3.2}$$

where $B_n(0) \triangleq I$ and the $B_n(j)$ for $1 \leq j \leq n$ are linear operators on $\mathscr{H}$. Because of the nesting property $\mathscr{K}_n \subset \mathscr{K}_{n+1}$ and the fact that $\Psi_n(U)\mathscr{H} = \mathscr{K}_n \ominus \mathscr{K}_{n-1}$, the $\Psi_n(U)\mathscr{H}$ for $n \geq 0$ form an orthogonal sequence of subspaces satisfying

$$\mathscr{K} = \bigoplus_0^\infty \Psi_n(U)\mathscr{H}\ .$$

Hence $\{\Psi_n(U)\mathscr{H}\}_0^\infty$ is a sequence of operators mapping $\mathscr{H}$ into $\mathscr{K}$ with mutually orthogonal ranges. In particular,

$$\Psi_m(U)^*\ \Psi_n(U) = \delta_{m,n}\ \epsilon_n^2 \qquad (m,n \geq 0) \tag{3.3}$$

where ϵ_n^2 is a positive operator on $\mathscr{H}$, and $\delta_{m,n} = 1$ for $m = n$ and zero otherwise.

Next, we will show that ϵ_n^2 and $\{B_n(j)\}_1^n$ can be determined via the multidimensional Levinson algorithm [MVK]. To this end, note that the nesting property of the $\mathscr{K}_j$ implies that $\Psi_n(U)\mathscr{H}$ is orthogonal to $U^jC^*\mathscr{H}$ for all $j < n$. Therefore

$$CU^{*j} \sum_0^n U^{n-i}\ C^*B_n(i) = 0 \qquad (\text{for } 0 \leq j < n)\ . \tag{3.4}$$

The nesting property with (3.2) and (3.3) also yields

$$(\epsilon_n^2 h,g) = (\Psi_n(U)h,\Psi_n(U)g) = (\Psi_n(U)h,U^nC^*B_n(0)g) = (CU^{*n}\sum_0^n U^{n-j}C^*B_n(j)h,g)$$

for all g and h in $\mathscr{H}$, so that

$$\sum_0^n CU^{*n}U^{n-j}C^*B_n(j) = \epsilon_n^2 \,. \tag{3.5}$$

Substituting (2.8) into (3.4) and (3.5), and noting that U* is a left inverse of U produces the usual multidimensional Levinson system for determining the $\{B_n(j)\}_1^n$:

$$\begin{bmatrix} R_0 & R_1^* & \cdots & R_n^* \\ R_1 & R_0 & \cdots & R_{n-1}^* \\ \vdots & \vdots & & \vdots \\ R_n & R_{n-1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} I \\ B_n(1) \\ \vdots \\ B_n(n) \end{bmatrix} = \begin{bmatrix} \epsilon_n^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.6}$$

where $R_j \triangleq R_X(j)$. Throughout this paper we will assume that the block Toeplitz matrix in (3.6) is invertible. Then obviously the $B_n(j)$ for $1 \leq j \leq n$ are uniquely determined linear operators on $\mathscr{H}$, and ϵ_n^2 is an invertible positive operator. The solution of (3.6) can be computed recursively via the well-known multidimensional Levinson algorithm. We now proceed to show how these $B_n(j)$ along with ϵ_n^2 determine the eigenvalues and corresponding eigenvectors of U*.

To this end, define the n^th^ Levinson operator mapping $\mathscr{H}$ onto $\mathscr{H}_n \ominus \mathscr{H}_{n-1}$ by

$$\Phi_n(U) = \Psi_n(U)\epsilon_n^{-1} = \sum_0^n U^{n-j}C^*B_n(j)\epsilon_n^{-1} = \sum_0^n U^{n-j}C^*\hat{B}_n(j) \tag{3.7}$$

where $\hat{B}_n(j) = B_n(j)\,\epsilon_n^{-1}$ are the normalized matrix-valued Levinson coefficients which can be computed recursively [MVK]. From (3.3) it follows that $\{\Phi_n(U)\}_0^\infty$ is a sequence of operators from $\mathscr{H}$ into $\mathscr{H}$ satisfying

$$\Phi_m(U)^*\Phi_n(U) = \delta_{m,n}I \quad (m,n \geq 0) \;\text{ and }\; \mathscr{H} = \bigoplus_0^\infty \Phi_n(U)\mathscr{H} \,. \tag{3.8}$$

Similarly, define the n^th^ matrix-valued Levinson polynomial in $z \epsilon C$ by

$$\phi_n(z) = \sum_0^n z^{n-j}\hat{B}_n(j) \qquad (n \geq 0) \,. \tag{3.9}$$

Notice that $\Phi_n(U)$ is *not* obtained from $\phi_n(z)$ by formally replacing z by U.

The following lemma and two theorems generalize those given in [FFS] for the case when $\mathscr{H}$ has dimension equal to one. They provide a complete characterization of the eigenvalues and corresponding eigenvectors of U*.

LEMMA 1. *If* $U^*y = \bar{\lambda}y$ *then*

$$\sum_0^\infty \|\phi_n(\lambda)^*Cy\|^2 < \infty \quad \textit{and} \quad y = \sum_0^\infty \Phi_n(U)\phi_n(\lambda)^*Cy \,. \tag{3.10}$$

PROOF. By virtue of (3.8) y has a Fourier series expansion of the form $y = \sum_0^\infty \Phi_n(U)h_n$ for some sequence $\{h_n\}$ in $\mathscr{H}$. From (3.8) this implies $h_n = \Phi_n(U)^*y$, so that

$$y = \sum_0^\infty \Phi_n(U)\Phi_n(U)^*y = \sum_0^\infty \Phi_n(U)\,[\sum_0^n \hat{B}_n^*(j)CU^{*n-j}]y =$$

$$\sum_0^\infty \Phi_n(U)\,[\sum_0^n \hat{B}_n^*(j)C\bar{\lambda}^{n-j}y] = \sum_0^\infty \Phi_n(U)\phi_n(\lambda)^*Cy \,.$$

Thus, by virtue of (3.8) we have

$$\sum_0^\infty \|\phi_n(\lambda)^*Cy\|^2 = \sum_0^\infty \|\Phi_n(U)\phi_n(\lambda)^*Cy\|^2 = \|y\|^2 < \infty \,.$$

This completes the proof.

At this point it is convenient to introduce the operator $X(\lambda)$ defined by

$$X(\lambda)h \triangleq \sum_{0}^{\infty} \Phi_n(U)\phi_n(\lambda)^* h \tag{3.11}$$

on the vectors h in $\mathscr{H}$ satisfying $\sum_{0}^{\infty} \|\phi_n(\lambda)^* h\|^2 < \infty$. As a result of (3.8) the series defining $X(\lambda)h$ is strongly convergent in $\mathscr{H}$.

The following two theorems provide a more complete and explicit characterization of the eigenvalues and eigenvectors of U^*. The first considers eigenvalues of U^* in D, and the second addresses eigenvalues on the unit circle.

THEOREM 1. *If* λ *is in* D *then* $\bar{\lambda}$ *is an eigenvalue of* U^* *and* $X(\lambda)$ *is a bounded linear operator from* $\mathscr{H}$ *into* $\mathscr{H}$. *In this case* $U^*y = \bar{\lambda}y$ *if and only if* $y = X(\lambda)h$ *where* h *is in* $\mathscr{H}$. *Moreover, if* γ *and* λ *are in* D *then*

$$\sum_{0}^{\infty} \phi_n(\gamma)\phi_n(\lambda)^* = (1 - \bar{\lambda}\gamma)^{-1} \left[\Theta(\lambda)^*\Theta(\gamma)\right]^{-1}. \tag{3.12}$$

PROOF. It is straightforward to show that any λ in D is an eigenvalue of multiplicity p for S^* (see [H1]). It follows that $\bar{\lambda}$ is an eigenvalue for $U^* = W^* \oplus S^*$. In this case, any y is a corresponding eigenvector for U^* if and only if y has the form

$$y = 0 \oplus f_\lambda \quad \text{where} \quad f_\lambda(e^{i\omega}) = (1 - \bar{\lambda}e^{i\omega})^{-1} f \quad \text{for f in } \mathscr{H}.$$

Also, for this y :

$$Cy = C_0 f_\lambda = (2\pi)^{-1} \int_{0}^{2\pi} (1 - \bar{\lambda}e^{i\omega})^{-1}\Theta(e^{i\omega})^* f \, d\omega = \Theta(\lambda)^* f .$$

This and the invertibility of $\Theta(\lambda)$ shows that the range of C on the eigenvectors of U^* corresponding to $\bar{\lambda}$ is the whole space $\mathscr{H}$. From Lemma 1 it follows that $X(\lambda)$ is defined on all of $\mathscr{H}$, and

$$X(\lambda)\Theta(\lambda)^* f = X(\lambda)Cy = y = 0 \oplus f_\lambda . \tag{3.13}$$

Since $\Theta(\lambda)$ is invertible we also deduce that $X(\lambda)\mathscr{H}$ is the eigenspace corresponding to the eigenvalue $\bar{\lambda}$ for U^*. This completes the proof of the first two parts.

To verify (3.12) consider any f and g in $\mathscr{H}$. Then from (3.13) we have

$$\begin{gathered}(1 - \bar{\lambda}\gamma)^{-1}(f,g) = (f_\lambda, g_\gamma) = (X(\lambda)\Theta(\lambda)^* f, X(\lambda)\Theta(\gamma)^* g) = \\ (\sum_{0}^{\infty} \Phi_n(U)\phi_n(\lambda)^*\Theta(\lambda)^* f, \sum_{0}^{\infty} \Phi_m(U)\phi_m(\gamma)^*\Theta(\gamma)^* g) = \\ (\Theta(\gamma) \sum_{0}^{\infty} \phi_n(\gamma)\phi_n(\lambda)^*\Theta(\lambda)^* f, g) .\end{gathered} \tag{3.14}$$

Since (3.14) holds for all f and g in $\mathscr{H}$ we have verified (3.12). This completes the proof.

We remark that if $\mathscr{H}$ is one-dimensional then (3.12) reduces to equation (12.3.17) in [S].

The following theorem is the main result of this paper. It characterizes the eigenvectors of U^* corresponding to eigenvalues on the unit circle. Since S^* has no eigenvalues on the unit circle these are precisely the eigenvalues of W^*, given by $\{e^{\pm i\omega_m}\}_1^k$. Recall that the set of signal frequencies is $\Omega = \{\omega_m\}_1^k$ and that $\{A_m\}_1^k$ is the corresponding set of amplitude matrices.

THEOREM 2. *Let* $|\lambda| = 1$. *Then* $\bar{\lambda} = e^{-i\omega_m}$ *is the* $(2m-1)^{th}$ *eigenvalue of* U^* *and* h *is in* $R(A_m)$ *for some* m $(1 \leq m \leq k)$, *if and only if*

$$\|X(\lambda)h\|^2 = \sum_{0}^{\infty} \|\phi_n(\lambda)^* h\|^2 < \infty . \tag{3.15}$$

Moreover, $U^*y = \bar{\lambda}y$ *if and only if* $y = X(\lambda)h$ *where* $\lambda = e^{i\omega_m}$ *and* h *is in* $R(A_m)$ *for some* m $(1 \leq m \leq k)$. *In this case*

$$A_m\Big(\sum_0^\infty \phi_n(\lambda)\phi_n(\lambda)^*\Big)A_m = 4\,P_m \tag{3.16}$$

where P_m *denotes the orthogonal projection onto* $R(A_m)$.

PROOF. Let $\bar\lambda = e^{-i\omega_m}$ and let f in $R(A_m)$ be arbitrary. We associate with f the eigenvector $y_\lambda(f) = y_u \oplus 0$ which has f appearing in the $(2m-1)^{\text{th}}$ position in y_u. Notice that all eigenvectors corresponding to the eigenvalue $\bar\lambda$ of U^* are of this form. By Lemma 1 and the definition of W and C_u we have

$$y_\lambda(f) = X(\lambda)Cy_\lambda(f) = X(\lambda)C_u y_u = \frac{1}{2}X(\lambda)A_m f\ . \tag{3.17}$$

If h is in $R(A_m)$ then $h = (1/2)\,A_m f$ for some f in $R(A_m)$. Substituting this into (3.17) gives $y \triangleq y_\lambda(f) = X(\lambda)h$. Now (3.15) follows from $\|y\| = \|y_u\| = \|f\| < \infty$ and (3.8).

On the other hand let h in $\mathscr{H}$ satisfy (3.15). Then $y = X(\lambda)h$ is defined and by Theorem 1 for all μ in $D \cup \{\lambda\}$, g in $\mathscr{H}$ and $j \geq 0$ we have

$$\begin{aligned}(U^*X(\mu)h,\Phi_j(U)g) = (X(\mu)h,U\Phi_j(U)g) = \sum_0^\infty (\Phi_n(U)\phi_n(\mu)^*h,U\Phi_j(U)g) = \\ \sum_0^{j+1} (\Phi_n(U)\phi_n(\mu)^*h,U\Phi_j(U)g)\ .\end{aligned} \tag{3.18}$$

The second equality follows from (3.8), while the last equality is a consequence of $U\Phi_j(U)h$ being in $\mathscr{H}_{j+1}$ and the nesting property of the $\mathscr{H}_i$. If μ is in D then from Theorem 1 it follows that $X(\mu)h$ is an eigenvector for U^*. In this case we have

$$(\bar\mu\phi_j(\mu)^*h,g) = (\bar\mu X(\mu)h,\Phi_j(U)g) = (U^*X(\mu)h,\Phi_j(U)g) \qquad (\mu \epsilon D)\ .$$

Substituting this into (3.18) yields

$$(\bar\mu\phi_j(\mu)^*h,g) = \sum_0^{j+1} (\Phi_n(U)\phi_n(\mu)^*h,U\Phi_j(U)g)\ . \tag{3.19}$$

Since (3.19) can be viewed as a polynomial equation in $\bar\mu$ which holds for all μ in D, it will also hold for all complex μ, and in particular for $\mu = \lambda = e^{i\omega_m}$. Therefore by (3.18) and (3.19) we have

$$\begin{aligned}(\bar\lambda X(\lambda)h,\Phi_j(U)g) = (\bar\lambda\Phi_j(U)\phi_j(\lambda)^*h,\Phi_j(U)g) = (\bar\lambda\Phi_j(\lambda)^*h,g) = \\ \sum_0^{j+1} (\Phi_n(U)\Phi_n(\lambda)^*h,U\Phi_j(U)g) = (U^*X(\lambda)h,\Phi_j(U)g)\ .\end{aligned} \tag{3.20}$$

Since (3.20) holds for all $j \geq 0$ the relation (3.8) and (3.20) imply $U^*\,X(\lambda)h = \bar\lambda\,X(\lambda)h$. Because $X(\lambda)h$ is an eigenvector for U^* the first part of the proof shows that $y = X(\lambda)h = y_\lambda(f)$ for some f in $\mathscr{H}$. By Lemma 1 then $X(\lambda)h = X(\lambda)Cy = X(\lambda)A_m\,(1/2)f$. Using $g = h - (1/2)\,A_m f$ and $\Phi_0(U) = I$, we have

$$0 = \|X(\lambda)g\|^2 = \sum_0^\infty \|\phi_n(\lambda)^*g\|^2 \geq \|\phi_0(\lambda)^*g\|^2 = \|g\|^2 \ \geq 0\ ,$$

implying that $h = (1/2)\,A_m f$. Thus h is in $R(A_m)$. This completes the proof of the first part.

Finally, to verify (3.16) let g in $\mathscr{H}$ be arbitrary and let $f = P_m g$. Then (3.8) and (3.17) imply

$$\begin{aligned}(P_m g,g) = \|f\|^2 = \|y_\lambda(f)\|^2 = \frac{1}{4}\Big(A_m \sum_0^\infty \phi_n(\lambda)\phi_n(\lambda)^*A_m f,f\Big) = \\ \frac{1}{4}\Big(A_m \sum_0^\infty \phi_n(\lambda)\phi_n(\lambda)^*A_m g,g\Big)\ .\end{aligned} \tag{3.21}$$

Since (3.21) holds for any g in $\mathscr{H}$ we easily infer (3.16) by polarization. This completes the proof.

4. THE ALGORITHM

Theorem 2 provides the basis for a recursive algorithm for estimating the signal frequencies $\{\omega_m\}_1^k = \Omega$ and the corresponding set of amplitude matrices $\{A_m\}_1^k$. Before we describe the algorithm in detail recall that the multidimensional Levinson algorithm utilizes the pair of update equations:

$$\phi_{n+1}(\lambda) = [\lambda\phi_n(\lambda) - \alpha_n(\lambda)\rho_{n+1}](I - \rho_{n+1}^*\rho_{n+1})^{-1/2}$$

$$\alpha_{n+1}(\lambda) = [\alpha_n(\lambda) - \lambda\phi_n(\lambda)\rho_{n+1}^*]\ (I - \rho_{n+1}\rho_{n+1}^*)^{-1/2}$$

where $\alpha_n(\lambda)$ is the n^{th} backward prediction Levinson polynomial and ρ_n is a strict contraction. In particular, the operator ρ_n is called the n^{th} *reflection coefficient* for the correlation sequence $\{R_X(j)\}$. For further details see [MVK].

Our algorithm for testing whether or not ω is in Ω will rely on the convergence property of the sequence

$$M_n(\omega) = \sum_0^n \phi_j(e^{i\omega})\phi_j(e^{i\omega})^* \qquad (n \geq 0)\,. \tag{4.1}$$

Clearly, $M_n(\omega)$ may be computed recursively via

$$M_n(\omega) = M_{n-1}(\omega) + \phi_n(e^{i\omega})\ \phi_n(e^{i\omega})^*.$$

The following corollary provides an alternative procedure.

COROLLARY. *Let* λ *and* γ *be in* D. *Then*

$$\sum_0^n \phi_j(\gamma)\phi_j(\lambda)^* = (1 - \bar{\lambda}\gamma)^{-1}[\alpha_n(\gamma)\alpha_n(\lambda)^* - \bar{\lambda}\gamma\phi_n(\gamma)\phi_n(\lambda)^*] \qquad (\lambda,\gamma\epsilon D)\,. \tag{4.2}$$

For $\gamma = \lambda = e^{i\omega}$ *this partial sum becomes*

$$M_n(\omega) = \sum_0^n \phi_j(\lambda)\phi_j(\lambda)^* = \bar{\lambda}[\phi_n(\lambda)\phi_n'(\lambda)^* - \alpha_n(\lambda)\alpha_n'(\lambda)^*] - \phi_n(\lambda)\phi_n(\lambda)^* \tag{4.3}$$

where $'$ *denotes* $d(.)/d\gamma|_{\gamma=\lambda}$ *and* $d\bar{\gamma}/d\gamma \triangleq -1$.

PROOF. From the Levinson system (3.6) it is clear that the Levinson polynomials $\{\phi_j(\lambda),\alpha_j(\lambda)\}_0^n$ can be computed from $\{R_X(j)\}_0^n$. Denote the maximum entropy extension of this sequence by $\{\tilde{R}_X(j)\}_0^\infty$. The reflection coefficients $\tilde{\rho}_j$ for this sequence are given by $\tilde{\rho}_j = \rho_j$ for $0 \leq j \leq n$ and $\tilde{\rho}_j = 0$ for $j > n$. Obviously, the Levinson polynomials associated with this extension are

$$\tilde{\phi}_j(\lambda) = \begin{cases} \phi_j(\lambda) & (0 \leq j \leq n) \\ \lambda^{j-n}\phi_n(\lambda) & (j > n) \end{cases} \quad \text{and} \quad \tilde{\alpha}_j(\lambda) = \begin{cases} \alpha_j(\lambda) & (0 \leq j \leq n) \\ \alpha_n(\lambda) & (j > n) \end{cases} \tag{4.4}$$

This implies that the outer (or minimum phase) spectral factor associated with the maximum entropy extension is

$$\tilde{\Theta}^{(n)}(\lambda) = \alpha_n^{-1}(\lambda)\,. \tag{4.5}$$

As a result, the application of Theorem 1 yields

$$(1 - \bar{\lambda}\gamma)^{-1}\alpha_n(\gamma)\alpha_n(\lambda)^* = \sum_0^\infty \tilde{\phi}_j(\gamma)\tilde{\phi}_j(\lambda)^* = \sum_0^n \phi_j(\gamma)\phi_j(\lambda)^* +$$

$$\sum_1^\infty (\gamma\bar{\lambda})^j\phi_n(\gamma)\phi_n(\lambda)^* = \sum_0^n \phi_j(\gamma)\phi_j(\lambda)^* + \gamma\bar{\lambda}(1 - \bar{\lambda}\gamma)^{-1}\phi_n(\gamma)\phi_n(\lambda)^*\,. \tag{4.6}$$

This proves (4.2).

Equation (4.3) follows directly from application of L'Hopital's rule to (4.2). This completes the proof.

Equation (4.2) is identical in form to the Christoffel-Darboux formula for scalar-valued orthogonal polynomials [S]. It is well known and has been derived using a number of different approaches, including reproducing kernels [KVM] and canonical recurrences [DGK]. Our proof incorporating the maximum entropy extension of $R_X(n)$ was given to demonstrate the utility of (3.12) in Theorem 1.

We remark that when evaluated on the unit circle, the expression (4.3) for $M_n(\omega)$ is notably different than the form obtained in [FFS] for the scalar case where the backward prediction polynomial $\alpha_n(\lambda) = \lambda^n \phi_n(\lambda^{-1})$ is simply the reverse of $\phi_n(\lambda)$.

We now present a procedure for estimating the set of signal frequencies and corresponding amplitude matrices.

PROCEDURE. As in the one-dimensional setting [FFS], the first step at the n^th^ stage is to compute the n^th^ degree Levinson polynomials. Having these, $M_n(\omega)$ can be efficiently computed via a fast Fourier transform algorithm. From Theorem 2 it follows that ω is contained in the set of signal frequencies Ω if and only if $M_n(\omega)$ is bounded on $R(A_m)$ for some m ($1 \leq m \leq k$). To use this fact we need some supplementary properties of $M_n(\omega)$. Let $\lambda_{1,n}(\omega) \leq \lambda_{2,n}(\omega) \leq \ldots \leq \lambda_{p,n}(\omega)$ denote the eigenvalues of $M_n(\omega)$. Since

$$M_{n+1}(\omega) = M_n(\omega) + \phi_{n+1}(e^{i\omega})\phi_{n+1}(e^{i\omega})^* \geq M_n(\omega)$$

each sequence $\{\lambda_{j,n}(\omega)\}_{n=1}^{\infty}$ for $1 \leq j \leq p$ is nondecreasing with $\lambda_{j,0}(\omega) = 1$. Let $\lambda_{j,\infty}(\omega) = \lim_{n\to\infty} \lambda_{j,n}(\omega)$. This sets the stage for the following result.

LEMMA 2. *The frequency ω is in Ω if and only if $\lambda_{1,\infty}(\omega)$ is finite. If $\omega = \omega_m$ is the m^th^ frequency in Ω then the rank of P_m is the largest r ($1 \leq r \leq p$) such that $\lambda_{r,\infty}(\omega)$ is finite. In this case P_m is the limit of $E_n(\omega_m)$ as $n \to \infty$ where $E_n(\omega_m)$ is the orthogonal projection onto the space generated by the eigenvectors corresponding to the eigenvalues $\lambda_{1,n}(\omega_m), \ldots, \lambda_{r,n}(\omega_m)$.*

PROOF. Let $\omega = \omega_n$ be in Ω and let R equal the rank of A_m. The minimax theorem [H2] for the R^th^ eigenvalue $\lambda_{R,n} \triangleq \lambda_{R,n}(\omega_m)$ of $M_n \triangleq M_n(\omega_m)$ implies that

$$\begin{aligned}\lambda_{R,n} \leq \max\{(M_n A_m h, A_m h) : \|A_m h\| = 1\} \leq 4\max\{\|P_m h\|^2 : \|A_m h\| = 1\} = \\ 4\max\{\|A_m^{-res} A_m h\|^2 : \|A_m h\| = 1\} \leq 4\|A_m^{-res}\|^2 < \infty\end{aligned} \tag{4.7}$$

where -res denotes the restricted inverse of A_m on $R(A_m)$. The second in inequality in (4.7) follows from (3.16) and the fact that the M_n's are increasing. Since the $\lambda_{R,n}$ are increasing, (4.7) implies that $\lambda_{R,\infty}$ is finite, hence $r \geq R$. In particular, the ordering of the eigenvalues shows that $\lambda_{1,\infty}$ is finite.

If R = p then r = R. If R < p let F_n denote the orthogonal projection onto the space generated by the eigenvectors corresponding to $\lambda_{1,n}, \ldots, \lambda_{R,n}, \lambda_{R+1,n}$. Since the rank of F_n is strictly greater than the rank of P_m there exists a unit vector h_n in $F_n\mathscr{H}$ which is orthogonal to $P_m\mathscr{H}$. Using the fact that the M_n's are increasing then for any $n_0 < n$ we have

$$(M_{n_0} h_n, h_n) \leq (M_n h_n, h_n) \leq \lambda_{R+1,n} < \lambda_{R+1,\infty}\,. \tag{4.8}$$

By choosing the appropriate subsequence, without loss of generality we can assume that $h_n \to h$ as $n \to \infty$. It follows that

$$(M_{n_0} h, h) \leq \lambda_{R+1,\infty} \quad \text{and} \quad P_m h = 0.$$

Letting $n_0 \to \infty$ and applying Theorem 2 we infer that $\lambda_{R+1,\infty}$ is infinite. Thus R = r. This proves the second statement in the lemma.

To finish the proof of the first statement suppose ω is not in Ω. Then upon replacing P_m by 0 and setting R = 0 in (4.8) a similar argument shows that $\lambda_{1,\infty}$ is infinite.

It remains to prove that E_n converges to P_m. If R = p then $P_m = E_m = I$, the identity on $\mathscr{H}$. So without loss of generality we can assume R < p. In this case (4.7) for $g = P_m g$ yields

$$4\|A^{-res}\|^2\|g\|^2 \geq (M_n g, g) \geq (M_n(I - E_n)g, (I - E_n)g) \geq$$

$$\lambda_{R+1,n}\|(I - E_n)g\|^2 \qquad (\text{for all } g = P_m g)$$

Since $\lambda_{R+1,\infty}$ is infinite we must have $E_n P_m \to P_m$ as $n \to \infty$. Because the rank of E_n equals the rank of P_m we deduce that $E_n \to P_m$. This completes the proof.

By virtue of Lemma 2 identification of Ω is based on the following test:

$$\begin{aligned} \lambda_{1,\infty}(\omega) &< \infty \text{ if and only if } \omega \in \Omega \\ \lambda_{1,\infty}(\omega) &= \infty \text{ if and only if } \omega \in \Omega\,. \end{aligned} \tag{4.9}$$

Since $\lambda_1(n)$ can be efficiently computed via the power method [F], this and (4.9) allows us to readily estimate the set of signal frequencies Ω.

Having found Ω, the next step is to determine each A_m. If the trace of $M_n(\omega_m)$ appears to converge for large n then we take A_m to be full rank, $P_m = I$ and

$$A_m \cong \left[M_n(\omega_m)\right]^{-\frac{1}{2}}\,. \tag{4.10}$$

If not, we must find the greatest integer r $(1 \leq r < p)$ for which $\lambda_{r,n}(\omega_m)$ appears to converge for sufficiently large n. Using this value of n we determine the orthogonal projection $E_n(\omega_m)$ defined in Lemma 2. We then take $P_m \cong E_n(\omega_m)$ and

$$A_m \cong \left[M_n(\omega_m)\right]^{-res/2} \tag{4.11}$$

where res denotes the restricted inverse of $M_n(\omega_n)$ corresponding to the range of $E_n(\omega_m)$.

In the one-dimensional setting $(k = 1)$ Lemma 2 reduces to

$$\lim_{n\to\infty} M_n^{-1}(\omega) = \begin{cases} A_m^2/4 & \text{for } \omega = \omega_m \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

where $M_n(\omega) = \sum_{0}^{n} |\phi_j(e^{i\omega})|^2$. Here $A_m^2/2$ is power of the m^th^ sinusoid at frequency ω_m as shown in [FFS]. In this case $M_n(\omega)$ reduces to the maximum likelihood spectral estimate developed by Capon [C]. Related results have also been obtained by Grenander and Szego [GS].

ACKNOWLEDGEMENTS

This research was supported in part by the Research Fund of Indiana University and the National Science Foundation No. ECS 8419354.

REFERENCES

[C] Capon, J., "High-resolution frequency-wavenumber spectrum analysis", *Proceedings of the IEEE,* **57**, 1408-1418, 1969.

[DGK] Delsarte, P., Genin, Y. and Kamp, Y., "Orthogonal polynomial matrices on the unit circle", *IEEE Trans. on Circuits and Systems,* **CAS-25**, 3, 149-160, 1978.

[FFS] Foias, C. Frazho, A. E. and Sherman, P. J., "An algorithm for estimating the amplitude and frequency components of harmonic signals in noise", to be presented at *International Symp. on Signal Processing and Appl.,* Brisbane, Australia, August 1987.

[F] Franklin, J., *Matrix Theory,* Prentice-Hall, 1968.

[GS] Grenander, U. and Szego, G., *Toeplitz Forms and Their Applications,* Chelsea Publ., New York, 1958.

[H1] Halmos, P. R. *A Hilbert Space Problem Book,* Springer-Verlag, 1982.

[H2] Halmos, P. R., *Finite-Dimensional Vector Spaces,* D. Van Nostrand Co., Princeton, 1958.

[K] Kailath, T., "A view of three decades of linear filtering theory", *IEEE Trans. on Information Theory,* **IT-20,** 145-181, 1974.

[KVM] Kailath, T., Vieira, A. and Morf, M., "Inverses of Toeplitz operators, innovations and orthogonal polynomials", *IEEE Conf. on Decision & Control,* 749-754, Dec. 1975.

[MVK] Morf, M., Vieira, A. and Kailath, T., "Covariance characterization by partial autocorrelation matrices", *Annals of Stats.,* **6**, 3, 643-648, 1978.

[MVLK] Morf, M., Vieira, A., Lee, D. and Kailath, T., "Recursive multichannel maximum entropy spectral estimation", *IEEE Trans. on Geoscience and Electronics,* GE - 16, 85-94, 1978.

[S] Szego, G., *Orthogonal Polynomials,* AMS, Providence, RI, 1975.

[SF] Sz.-Nagy, B. and Foias, C., *Harmonic Analysis of Operators on Hilbert Space,* North Holland, 1970.

ON THE OPTIMUM DELAY IN FREQUENCY-SAMPLING ALE

SEN M. KUO and MICHAEL A. RODRIGUEZ

Dept. of Elec. Engr.
Northern Illinois U.
DeKalb, IL 60115

Technical Center
R. R. Donnelley
Lisle, IL 60532

ABSTRACT

The optimum selection of the decorrelation delay for each frequency-sampling adaptive line enhancer (FSALE) module is presented. It is shown that the optimum delay for each module of FSALE is a function of the filter order N and module number k. Therefore, no a **priori** knowledge of sinusoidal line frequency is required to choose optimum delay for each module and this is especially advantageous for multi-line cases.

SUMMARY

The adaptive line enhancer (ALE)[1] provides an efficient means for adaptively tracking the sinusoidal components of a received signal and separates these narrow-band signals from broad-band noise. This technique has been shown to be effective in practical applications when there is insufficient a priori knowledge of the signal and noise parameters. The optimization of delay to minimize the mean square error for the special case of a single sinusoid in noise has been reported [2]. This optimal delay depends on the input frequency of sinusoidal which is unknown.

The block diagram of an FSALE [3] module is illustrated in Figure 1. The FSF is the k-th module modified frequency-sampling filter with order N. The key feature is the use of FSF to generate a set of "desired" signals from the input and a corresponding set of two-weight adaptive filters to gate line signals to the output. The delay D causes decorrelation between the noise component of the input signal x(i) and desired signal d(i). It can be shown that the noise components at the summing junction become decorrelated by choosing the delay to satisfy

$$D > 1 \tag{1}$$

In the FSALE scheme, we form independent error sequences for each module. Therefore, each module can have its own fixed optimum delay which will minimize the average error variance at that module. It can be shown that the optimum delay D for the k-th module satisfies

$$2D - 1 = (nN) / k \tag{2}$$

where n is some positive integer and k = 1,2,..,N/4. This holds symmetrically for the (N/2)-k modules and the optimum delay must also satisfy D > 1.

Optimum delay is thus a function of module number k and filter order N. Therefore, no a priori knowledge of sinusoidal line frequency is required to choose D for each module. This is especially advantageous for multi-line enhancement. For any k it is natural to choose the smallest integer n so D will be as small as possible and still satisfy (1) and (2).

In conclusion, we shown that substantial improvements could be obtained by choosing an optimum delay for each FSALE module. Continued research is proposed to expand these optimum delays into other frequency domain ALE structures [4,5].

REFERENCES

[1] "Adaptive Noise Canceling: Principles and Applications," by B. Widrow, et al, Proc. IEEE, Dec. 1975.

[2] "Optimized Lattice-Form Adaptive Line Enhancers for a Sinusoid Signal in Broadband Noise," by V. Reddy, et al, IEEE Trans. on ASSP, June 1981.

[3] "Theory and Implementation of an Adaptive Frequency-Sampling Line Enhancer," by S. M. Kuo, Ph. D. Dissertation, Univ. of New Mexico, Dec. 1985.

[4] "Transform Domain LMS Algorithm," by S. S. Narayan et al., IEEE Trans. on ASSP, June 1983.

[5] "The Recursive Adaptive LMS Filter-A Line Enhancer Application and Analytical Model for the Mean Weight Behavior," by N. Bershad, et al., IEEE Trans. on ASSP, December 1980.

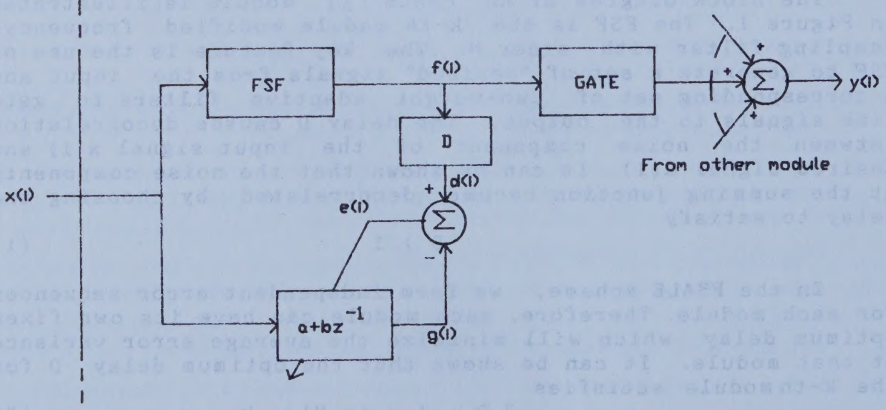


Fig. 1 A block diagram of FSALE, k-th module.

SYNTHESIS AND DESIGN OF STRUCTURALLY PASSIVE FULLY RECURSIVE HALF-PLANE 2-D DIGITAL FILTERS

Sankar Basu and K. Mahtre
Stevens Institute of Technology, Hoboken, NJ 07030.

ABSTRACT

Passive and lossless two-dimensional digital one-ports as well as two-ports of the fully recursive half-plane type are introduced and are characterized in terms of their transfer function descriptions. An algorithm for the structurally passive synthesis of filters having such recursive structure is then derived from these representation results as an extension of a recent Schur type algorithm for the synthesis of discrete lossless two-ports. Design methods for important practical cases when frequency response of the filter is required to have specific symmetries are also presented.

1. INTRODUCTION

Various recursive schemes have been proposed in the multidimensional (m-D) digital filter literature. Among these the most widely studied are the quarter plane, the asymmetric and the symmetric half-plane recursive scheme. More recently, motivated by needs for parallel processing of 2-D signals a scheme known as the fully recursive half-plane scheme has been proposed and a method of designing transfer functions of filters having this recursive structure has been outlined in [1]. The impulse responses of the class of filters just mentioned satisfies the characteristic property that the region of support is a half-plane and the filter is recursive in both horizontal and vertical direction. More specifically, the recursion equation describing the relation between the input x and output y of a filter of this type is given by:

$$\beta_0[\{y_n(m)\}] = -\sum_{i=1}^{L_D} \beta_i[\{y_{n-1}(m)\}] + \sum_{i=0}^{L_N} \alpha_i[\{x_{(n-1)}(m)\}] \tag{1.1}$$

where $x_n(m)$, $y_n(m)$ denote the n-th row of the input and the output signal, and the (row) operations $\alpha_i[.]$ and $\beta_i[.]$ respectively denote 1-D linear shift invariant convolution operations with the 1-D sequences $a_i(m)$ and $b_i(m)$. Considering the 2-D Z-transform of (1.1), and assuming that the operations $\alpha_i[.]$ and $\beta_i[.]$ are all rational we then have $T(z_1,z_2)$ in (1.2) for the transfer function of the filter, where $A_i(z_1)$, $B_i(z_1)$ are rational functions representing the row operations just mentioned.

$$T(z_1,z_2)= \sum_{i=0}^{L_N} A_i(z_1)z_2^i / \sum_{i=0}^{L_D} B_i(z_1)z_2^i \tag{1.2}$$

On the otherhand, it is now well known that an input-output specification is not enough for the successful operation of a digital filter but structural considerations need to be taken into account. The class of structurally passive filters variously known as the wave digital filters [2], orthogonal filters [3], when properly designed, are known to satisfy the properties of insensitivity to coefficient perturbation and non-linear arithmatic conditions resulting from overflow, finite precision arithmatic etc. Although much progress has been documented in the synthesis and design of 1-D structurally passive filters, methods for two and higher dimensions are still evolving. Synthesis methods for two and multidimensional wave digital filters, which are quarter plane type filters have been reported in [2], [4]. Quarter plane and asymmetric half-plane generalizations of 1-D lattice filters which are, in fact, structurally passive, have been discussed recently in the context of random field modeling in [5].

Following 1-D, in the present paper (pseudo)passive or (pseudo)lossless fully recursive half-plane 2-D digital filters are introduced and a method of their structurally passive synthesis and subsequently that of their design is discussed for the first time. The problem of synthesis of quarter plane causal (thus, including filters causal in a convex cone [6]) structurally passive multidimensional filters of the type mentioned above can be equivalently viewed as the classical network theoretic problem of synthesizing a lossless but otherwise arbitrarily prescribed multidimensional transfer function as an interconnection of elementary building blocks such as capacitors and inductors. This latter problem is completely unresolved in multidimensions (m>2), whereas in 2-D synthesis is feasible only in an unconstrained topological structure [6]. On the otherhand, it has been shown that if certain ladder-like constraints are imposed on the structure in which the filter is to be synthesized then the prescribed 2-D transfer function must satisfy further restrictions in addition to input-output losslessness [7]. Related other synthesis results [4], [2] in this context deal with important special cases when the multidimensional frequency response of the filter possesses certain symmetries. In contrast, the present work provides us with a synthesis of arbitrary lossless fully recursive half-plane 2-D filters. Additionally, unlike the quarter plane case referred to earlier the synthesis is obtained in a fixed predetermined structure potentially useful for practical implementation.

As in most passive or lossless filter design techniques our synthesis method proceeds by viewing the prescribed passive transfer function as being embedded into the transfer function of a lossless two-port. The synthesis of this fully recursive half-plane lossless two-port takes advantage of a recent algorithm for the design of structurally passive 1-D filters advanced by Rao and Kailath [8] as an extension of the celebrated Schur algorithm. Unlike all other methods known for the synthesis of 1-D continuous as well as discrete lossless two-ports including those available in the classical circuit theoretic literature, the algorithm of [8] enjoys the unique feature that given a transfer function associated with the lossless two-port the synthesis algorithm makes use of rational arithmatic operations only (i.e., nonrational arithmatic operations such as polynomial factorization is not required) [9]. The synthesis method for fully recursive half- plane filters to be presently described fully exploits this rational character of the 1-D algorithm in [8]. Although the details of the method differ nontrivially form 1-D due to considerations characteristic of multidimensional problems (e.g., those utilizing techniques from elementary algebraic curve theory [10]), the synthesis to be outlined can be considered, at least at a conceptual level, to be a generalization of the result in [8] to two-port transfer functions the coefficients of numerator and denominator polynomials of which belong to a field of rational functions (instead of the field of rational numbers). From a different perspective the present work can also be viewed as a generalization of 1-D Schur algorithm to 2-D fully recursive half-plane filters, thus making it possible to cast the present discussion in the closely related framework of modeling of stationary random fields and scattering theory.

A note regarding the stability of the filter is in order. The region of analyticity of the transfer function of our filter will be found to marginally differ from those previously considered in the 2-D half-plane literature [1]. This is primarily due to the fact that the results such as those in [1] are motivated by bounded-input-bounded- output considerations, whereas, in contrast, our results are driven by passivity considerations. The fact that this difference in consideration does indeed lead to diverging formulations of stability in multidimensions (m>1), but not in 1-D, is now known [11]. Thus, there is no contradiction between our stability results and those in the existing half-plane literature.

2. FULLY RECURSIVE HALF-PLANE PASSIVE SYSTEMS

By associating the (pseudo) energy $\Sigma\Sigma|x(n_1,n_2)|^2$ to the input $x(n_1, n_2)$ (and similarly to the output) of the system we first develop conditions necessary for the transfer function of a fully recursive half-plane filter to be passive in the sense that:

$$\Sigma\Sigma\ |y(n_1, n_2)|^2 \leq \Sigma\Sigma\ |x(n_1, n_2)|^2 \tag{2.1}$$

for any choice of square summable input sequence $x(n_1, n_2)$.

A consequence of passivity reflected on the transfer function $H(z_1, z_2)$ of the filter is that $|H(\omega_1,\omega_2)|\leq 1$ for all real two-tuples (ω_1,ω_2) except possibly for finitely many of them and $H(z_1,z_2)$ is bounded for all $|z_1| = 1$ and $|z_2| < 1$.

We next make the simplifying assumption that the operations $\alpha_i[.]$ and $\beta_i[.]$ in (1.1) are rational i.e., $A_i(z_1)$ and $B_i(z_1)$ are rational functions in z_1. Since the recursion equation for the fully recursive half-plane filter is given by (1.1) it is clear that the transfer function $H(z_1,z_2)$ is a rational function of z_2. Under the present assumption however, $H(z_1,z_2)$, becomes a rational function of both z_1 and z_2, and can be expressed as the ratio of two relatively prime polynomials $b(z_1,z_2)$ and $a(z_1,z_2)$ as:

$$H(z_1,z_2) = b(z_1,z_2)/a(z_1,z_2) \tag{2.2}$$

We now claim that for passive systems presently under consideration, the polynomial $a(z_1,z_2)$ in (2.2) cannot have infinitely many zeros on the distinguished boundary $|z_1| = |z_2| = 1$ of the unit bi-disc. For, if $a(z_{10},z_{20}) = 0$ for some $|z_{10}| = |z_{20}| = 1$ then in view of (2.2), in order for $|H(\omega_1,\omega_2)| \leq 1$ to be satisfied we would need $b(z_{10},z_{20}) = 0$ i.e., $a(z_1,z_2)$ and $b(z_1,z_2)$ would have a common zero on $|z_1|=|z_2|=1$. However, the presence of infinitely many such zeros would, in view of Bezout's theorem in algebraic curve theory [10], require that $b(z_1,z_2)$ and $a(z_1,z_2)$ have a common factor, which is hypothesized to be absent in (2.2).

A further consequence of passivity is that when the transfer function $H(z_1,z_2)$, is expressed in terms of ratio of two relatively prime polynomials $b(z_1,z_2)$ and $a(z_1,z_2)$ as in (2.2), it must be true that $a(z_1,z_2)$ is relatively prime with the polynomial $\hat{a}(z_1,z_2)$ defined as:

$$\hat{a}(z_1,z_2) = \tilde{a}(z_1,z_2)z_1^{d_1}z_2^{d_2}; \; \tilde{a}(z_1,z_2) = a^*(z_1^{*-1},z_2^{*-1}) \tag{2.3 a,b}$$

where d_1, d_2 are the partial degrees of a in z_1 and z_2, and * denotes complex conjugation. To show this let $g(z_1,z_2)$ be the g.c.d between $a(z_1,z_2)$ and $\hat{a}(z_1,z_2)$. Then as shown in [11] we must have $\hat{g}(z_1,z_2) = \gamma g(z_1,z_2)$ where γ = constant, $|\gamma|=1$; and $g(z_1,z_2)$ is either a constant or must have infinitely many zeros on $|z_1| = |z_2| = 1$. In the latter case, $a(z_1,z_2)$ must have infinitely many zeros on $|z_1| = |z_2| = 1$, which has been shown to be impossible in the last paragraph. Thus, $g(z_1,z_2)$ = constant, and $a(z_1,z_2)$ is relatively prime with $\hat{a}(z_1,z_2)$.

The essential features of the above discussion are summerized in the following result.
Property 2.1: The transfer function of a passive fully recursive half-plane filter, when expressed in irreducible rational form as in (2.2), satisfies the following two conditions: (i) $a(z_1,z_2) \neq 0$ for $|z_1| = 1$ and $|z_2| < 1$ i.e., $H(z_1,z_2)$ is analytic in $|z_1| = 1$ and $|z_2| < 1$. (ii) $a(z_1,z_2)$ and $\hat{a}(z_1,z_2)$ as defined in (2.3), do not have a common factor.

The condition (ii) in the above property can, in fact, be replaced by any one of the conditions expressed in the following.

Assertion 2.1: Let $a(z_1,z_2)$ be a polynomial in z_1 and z_2 such that $a(z_1,z_2) \neq 0$ for $|z_1| = 1$ and $|z_2| < 1$. Then the following conditions are all equivalent: (a) $a(z_1,z_2)$ does not have infinitely many zeros on the distinguished boundary $|z_1| = |z_2| = 1$ (b) $a(z_1,z_2)$ and $\hat{a}(z_1,z_2)$ are relatively prime polynomials (c) each irreducible factor of $a(z_1,z_2)$ has at least one zero in the domain $|z_1| = 1$, $|z_2| > 1$.

We also have the following important result.

Property 2.2a: If a rational function $H=H(z_1,z_2)$ as in (2.11) is such that $|H|\leq 1$ on $|z_1|=|z_2|=1$ except possibly at finite number of points where H is not well defined and if H satisfies the conditions expressed in property 2.1(i) (thus, if $H(z_1,z_2)$ is tranfer function of a passive fully recursive half plane filter) then $|H|\leq 1$ for all $|z_1|=1$ and $|z_2|<1$. Furthermore, if for some (z_{10},z_{20}) with $|z_{10}|=1, |z_{20}|<1$ $|H(z_{10},z_{20})|=1$ then $H(z_{10},z_2)$ is a constant independent of z_2. Assuming H to be nonconstant, the latter situation can arise for at most finitely many values of z_{10} (with $|z_{10}|=1$).

In fact, the following result in Property 2.2b can also be proved. This result shows that the polynomials of the type described in properties 2.1 (i) and 2.1 (ii) characterize denominator of irreducible rational functions satisfying the (half-plane) boundedness property: $|H|\leq 1$ for $|z_1|=1$, $|z_2|<1$.

Property 2.2b: If H is a nonconstant irreducible rational function as expressed in (2.2) and is such that $|H|\leq 1$ for $|z_1|=1, |z_2|<|$ then either a is a constant or satisfies properties 2.1 (i) and 2.1 (ii).

We next assume the filter to be (pseudo)lossless in the sense described earlier i.e., equations (2.1) is satisfied with equality. Consequently, from (2.1) we then have that for all 2-tuples (ω_1,ω_2) with the possible exception of finitely many (2.4a) holds true. Consequently, the rational transfer function $H(z_1,z_2)$ of a (pseudo)lossless fully recursive half-plane transfer function also satisfies (2.4b).

$$|H(\omega_1,\omega_2)| = 1 \qquad \tilde{H}(z_1,z_2)\, H(z_1,z_2) = 1 \tag{2.4a,b}$$

For convenience of further exposition the following terminology will be introduced. Any rational function $H(z_1,z_2)$ as expressed in (2.2) will be said to be a fully recursive half-plane all-pass function if $H(z_1,z_2)$ satisfies the conditions stated in property 2.1 and in equation (2.4b). Thus, transfer functions of (pseudo)lossless fully recursive half-plane filters are fully recursive half-plane all-pass functions.

A function $A(z_1,z_2)$ of two variables z_1, z_2, when expressible as a polynomial in z_2 with coefficients as rational functions in z_1 will be said to be a pseudopolynomial (in z_2). Thus, if $A(z_1,z_2)$ is a pseudopolynomial then

$$A(z_1,z_2) = \alpha_0(z_1) + \alpha_1(z_1)z_2 + \ldots + \alpha_{N_2}(z_1)z_2^{N_2} \tag{2.5}$$

where $\alpha_k(z_1)$'s are rational functions in z_1. With $A(z_1,z_2)$ as given in (2.5), where $\alpha_{N_2}(z_1)$ is not identically zero, the integer N_2 will also be denoted by $\deg_2 A$. Furthermore, the notation $\bar{A}(z_1,z_2)$ will be used to denote the pseudo- polynomial obtained from $A(z_1,z_2)$ as:

$$\bar{A}(z_1,z_2) = \tilde{A}(z_1,z_2)z_2^{N_2} \tag{2.6}$$

Two pseudopolynomials $B(z_1,z_2)$ and $C(z_1,z_2)$ are said to be coprime if there is no pseudopolynomial $D(z_1,z_2)$ actually involving z_2 such that $B(z_1,z_2) = D(z_1,z_2)\, B_1(z_1,z_2)$ and $C(z_1,z_2) = D(z_1,z_2)\, C_1(z_1,z_2)$ for some pseudopolynomials $B_1(z_1,z_2)$ and $C_1(z_1,z_2)$. The following property then holds true.

Property 2.3: Any fully recursive half-plane all-pass function $H(z_1,z_2)$ (thus, rational transfer function of (pseudo)lossless fully recursive half-plane filter) can be expressed as follows:

$$H(z_1,z_2) = -D(z_1)\,[A(z_1,z_2)/\bar{A}(z_1,z_2)] \tag{2.7}$$

where i) $A(z_1,z_2)$ is a pseudopolynomial

ii) $D(z_1) = z_1^N\gamma\,[d(z_1)/\hat{d}(z_1)]$, where $d(z_1)$ is a polynomial in z_1, γ is a constant of unit modulus and N = integer.

iii) the pseudopolynomials $A(z_1,z_2)$ and $\bar{A}(z_1,z_2)$ are coprime,

iv) $\bar{A}(z_1,z_2) \neq 0$ for all $|z_1|=1$, $|z_2|<1$.

Conversely, any rational function expressible as in (2.7) with (i), (ii), (iii) and (iv) in force is a fully recursive half-plane all-pass function.

3. FULLY RECURSIVE HALF-PLANE LOSSLESS TWO-PORTS

In this section we make use of the results of the previous section to characterize fully recursive half-plane lossless two-ports. In particular, a convenient representation for such two-ports analogous to the Belevitch canonical representation of continuous time 1-D lossless circuits of classical network theory [12] is developed. This representation is then subsequently used in section 4 to synthesize the filter in a specific structure.

A two-port of the type under consideration with $x_1(.)$, $x_2(.)$ as its input and $y_1(.)$, $y_2(.)$ as its output is lossless if (3.1) holds true for any finite (pseudo)energy inputs $x_1(n_1,n_2)$ and $x_2(n_1,n_2)$.

$$\Sigma\Sigma|y_1(n_1,n_2)|^2+\Sigma\Sigma|y_2(n_1,n_2)|^2 = \Sigma\Sigma|x_1(n_1,n_2)|^2+\Sigma\Sigma|x_2(n_1,n_2)|^2 \tag{3.1}$$

If $S = S(z_1,z_2) = [S_{ij}(z_1,z_2)]$ is the transfer function of the two-port then it can be shown that $S_{ij} = S_{ij}(z_1,z_2)$ for $i,j = 1,2$ satisfy Property 2.1.

Furthermore, by considering 2-D Parseval's theorem (3.1) can be made to yield (3.2).

$$|S(\omega_1,\omega_2)| = I_2, \quad \tilde{S}(z_1,z_2)\, S(z_1,z_2) = I_2 \tag{3.2a,b}$$

A (2×2) rational matrix $S = S(z_1,z_2)$ is said to be fully recursive half-plane lossless bounded if: (i) each entry of S satisfies the conditions expressed in properties 2.1 (i), (ii) and equation (3.2b) holds true.

Note that the transfer function of a fully recursive half- plane lossless two-port is necessarily of the above type. As a consequence of property 2.2, we then have the following important conclusion.

Proposition 3.1: Each entry of a fully recursive half-plane bounded matrix $S = [S_{ij}]$ satisfies $|S_{ij}| < 1$ for all $|z_1|=1$ and $|z_2|<1$.

Our next result concerns characterization of a fully recursive half-plane lossless bounded matrix S.

Property 3.1: Any fully recursive half-plane lossless bounded matrix $S=[S_{ij}]$ of a two-port can be represented in terms of three pseudopolynomials A, B and C as in (3.3) and (3.4), where $\bar{A}$ is nonzero in $|z_1|=1$, $|z_2|<1$; the pseudopolynomials A and $\bar{A}$ are coprime; and furthermore (3.5) hold true.

$$S_{11} = B/\bar{A}, \quad S_{22} = -D(\tilde{B}/\tilde{A}) \tag{3.3a,b}$$

$$S_{21} = C/\bar{A}, \quad S_{12} = D(\tilde{C}/\tilde{A}) \tag{3.4a,b}$$

$$A\tilde{A} = B\tilde{B} + C\tilde{C}, \quad \deg_2 B \le \deg_2 A \;;\; \deg_2 C \le \deg_2 A \tag{3.5a,b}$$

Conversely, any matrix, which admits of the above representation is fully recursive half-plane bounded.

A fully recursive half-plane lossless two-port can be alternatively described by means of a chain matrix $T = T(z_1,z_2)$. It can be easily shown that the following property characterizes the chain matrices of the type described above.

Property 3.1': The chain matrix $T = [T_{ij}]$ associated with a fully recursive half-plane two-port is lossless if and only if it can be expressed as

$$T_{11} = DA/C \;; \qquad T_{12} = B/C \tag{3.6a,b}$$

$$T_{21} = D\tilde{B}z_2^{n}A/C \;;\; T_{22} = \bar{A}/C \tag{3.7a,b}$$

where $n_A = \deg_2 A$, and A, B, C and D satisfies the same restrictions described in property 3.1. Any S or T as in the above propositions is said to be in standard form.

4. SYNTHESIS OF FULLY RECURSIVE HALF-PLANE LOSSLESS TWO-PORTS

A procedure for synthesizing fully recursive half-plane lossless two-ports exploiting the unique feature of the 1-D Rao-Kailath algorithm [8] that (in 1-D) given (polynomials) A, B, C only rational arithmetic operations are needed to be performed on the coefficients of A, B and C will be developed. The basic structure of the filter to be presently synthesized would thus be the same as in [8], whereas the elementary building blocks are certain 1-D two port sections to be referred to as the generalized Gray-Markel sections (GGM section) and z_2-type delays, each of which are fully recursive half-plane lossless.

A GGM section is a 1-D two port as shown in figure 4.1 where the 1-D transfer functions (assumed rational) k_1 and k_2 satisfy the relationship (4.1).

$$k_1\tilde{k}_1 + k_2\tilde{k}_2 = 1 \tag{4.1}$$

and are such that k_1 (and thus k_2 in view of (4.1)) satisfies $k_1\tilde{k}_1 = |k_1|^2 < 1$ everywhere on $|z_1| = 1$ with possible exception of finite number of z_1 for which we may have $|k_1|^2 = 1$.

We note that given any rational function k_1 of z_1 satisfying the above conditions it is always possible to find a rational function k_2 satisfying the same conditions as that of k_1 along with (4.1). (The role of k_1 and k_2 can obviously be interchanged in the present considerations). However, the k_2 so obtained is not unique unless further restrictions are imposed.

The transfer function matrix $S_G = S_G(z_1)$ associated with such a GGM section can be expressed as in (4.2a), whereas the corresponding chain matrix T is given in (4.2b).

$$S_G = \begin{bmatrix} k_1 & k_2 \\ -\tilde{k}_2 & -\tilde{k}_1 \end{bmatrix} \qquad T_G = (1/\tilde{k}_2)\begin{bmatrix} 1 & k_1 \\ \tilde{k}_1 & 1 \end{bmatrix} \tag{4.2a,b}$$

Since S_G in (4.2a) satisfies the representation described in Property 3.1 with $A = 1$, $B = k_1$, $C = k_2$ and $D = 1$ the GGM section is indeed a fully recursive half-plane lossless two-port.

To proceed with the synthesis of a prescribed S, as described in property 3.1, we first note that in view of Porposition 3.1 the rational function $k_1 = k_1(z_1) = S_{11}(z_1,0)$ satisfies $|k_1| < 1$ for all $|z_1| = 1$ with the possible exception of finite number of values of z_1 where $|k_1|=1$. Therefore, in view of the preceeding discussion k_1 defines a GGM section i.e., a k_2 can be found such that $|k_2|<1$ everywhere on $|z_1|=1$ with the possible exception of finite number of values of z_1, for which $|k_2|=1$ and that (4.1) is satisfied.

Step 1: The first step is to extract a GGM section with $k_1 = S_{11}(z_1,0)$ from prescribed S or T as shown in figure 4.2. Since a cascade connection of two two-ports amounts to multiplication of the corresponding chain matrices, the chain matrix of the remaining two-port is then $T' = T_G^{-1}T$. From (3.6), (3.7) and (4.2b) we can write:

$$T' = (1/Ck_2)\begin{bmatrix} D(A-k_1\tilde{B}z_2^{n_A}) & B-k_1\bar{A} \\ D(\tilde{B}z_2^{n_A}-\tilde{k}_1A) & \bar{A}-\tilde{k}_1B \end{bmatrix} \tag{4.3}$$

We next define the pseudopolynomials A', B', C' and the 1-D rational function D' as in (4.4) and (4.5) below, where $p=p(z_1)$ is the 1-D conjugate reciprocal [11] polynomial factor of largest degree present in the numerator of $\bar{A}-\tilde{k}_1B$, when expressed in irreducible rational form.

$$\tilde{p}A' = A(1-k_1\tilde{S}_{11}) = A-k_1\tilde{B}z_2^{n_A} \; ; \; pC' = Ck_2 \tag{4.4a,b}$$

$$pB' = \bar{A}(S_{11}-k_1) = B-k_1\bar{A} \quad ; \quad D' = D(\tilde{p}/p) \tag{4.5a,b}$$

We claim that $\deg_2 A' = \deg_2 A$. To prove this, clearly $\deg_2 A' \leq \deg_2 A$ and note that (4.4a) yields $\tilde{p}A'/\tilde{A} = 1-k_1S_{11}$, which implies that if $\deg_2 A' < \deg_2 A$ then for arbitrary z_1 we would have $\tilde{k}_1(z_1)S_{11}(z_1,0) = |S_{11}(z_1,0)|^2 = 1$. The impossibility of this latter situation has already been demonstrated. As a consequence of this we can write T′ as in (4.6) and (4.7), where $n_{A'} = \deg_2 A' = \deg_2 A$.

$$T'_{11} = D'A'/C' \; ; \; T'_{12} = B'/C' \tag{4.6a,b}$$

$$T'_{21} = D'\tilde{B}'z_2^{n_{A'}}/C' \;;\; T'_{22} = \bar{A}'/C' \qquad (4.7a,b)$$

We next claim that the pseudopolynomial $\bar{A}'$ satisfies the properties that $\bar{A}' \neq 0$ for $|z_1|=1$, $|z_2|<1$ and that $\bar{A}'$ is coprime with A'. To prove this we write $A' = A'_N/A'_D$ in irreducible rational form, and thus $\bar{A}' = (\hat{A}'_N/\hat{A}'_D).z_1^{\alpha}$, where α = integer and $\hat{A}'_N/A'_D$ is in irreducible rational form. Thus, from the definition of $\bar{A}'$ it follows that $\hat{A}'_N$ is devoid of conjugate reciprocal polynomial factors in z_1 only. If we assume for the prupose of a proof by contradiction that for some value of $z_1=z_{10}, z_2=z_{20}$ with $|z_{10}|=1$, $|z_{20}|<1$ we have $\bar{A}'=0$ i.e., $\hat{A}'_N=0$ then since $\hat{A}'_N$ cannot have a factor (z_1-z_{10}), by changing the value of z_1 from z_{10} along an arbitrarily small arc Γ_1 of the unit circle $|z_1|=1$ it would be possible to find a continuous [11] set (z_1,z_2) of zeros of $\hat{A}'_N$ i.e., also of $\bar{A}'$ with $z_1 \epsilon \Gamma_1 \subset \{z_1; |z_1|=1\}$ and $|z_2|<1$.

Also, since it follows from (4.4a) and $n_A=n_A'$ that $p\bar{A}'=\bar{A}(1-\tilde{k}_1S_{11})$ and $\bar{A}\neq 0$ in $|z_1|=1$, $|z_2|<1$ (cf. property 3.1) we would then have that for all $z_1 \epsilon \Gamma_1$ there exists some z_2 in $|z_2|<1$. such that $k_1\tilde{S}_{11}=1$. Since $|k_1|\leq 1, |S_{11}|\leq 1$ for all $|z_1|=1, |z_2|<1$ (cf. Proposition 3.1) the last conclusion implies that there exists z_2 in $|z_2|<1$ such that $|k_1|=|S_{11}(z_1,0)|=1$ and $|S_{11}(z_1,z_2)|=1$ for all $z_1 \epsilon \Gamma_1$. However, this last conclusion leads to a contradiction in view of Property 2.2a. Thus, $\bar{A}'\neq 0$ i.e., $\bar{A}'_N\neq 0$ for $|z_1|=1, |z_2|<1$.

Also, since due to Property 2.2a $|\tilde{k}_1|=|S_{11}(z_1,0)|=1$ is possible for at most finite number of values of z_1, for $|z_1|=|z_2|=1$ we have $|S_{11}|\leq 1$ and $\bar{A}\neq 0$ for at most finite number of exceptions on the distinguished boundary of the unit bi-disc, we conclude from $p\bar{A}'=\bar{A}(1-\tilde{k}_1S_{11})$ that $\bar{A}'$, thus $\hat{A}'_N$ may have at most finite number of zeros on $|z_1|=|z_2|=1$. Since as shown earlier $\bar{A}'_N\neq 0$ in $|z_1|=1, |z_2|<1$ it follows from Assertion 2.1 that A_N and $\hat{A}_N$ are relatively prime polynomials. Consequently, the pseudopolynomials A' and $\bar{A}'$ are relatively prime.

Finally, straightforward algebraic manipulation shows that the conditions analogous to (3.5) necessary for $T' = [T'_{ij}]$, as given in (4.6), (4.7), to be a fully recursive half-plane lossless two-port chain matrix are satisfied.

We further note that as a consequence of the choice $k_1=S_{11}(z_1,0)$ we have from (4.5a) that $B'(z_1,0) = 0$ for arbitrary z_1 i.e., the pseudopolynomial B' contains z_2 as a factor. Also, if C contains a pseudopolynomial factor z_2 then so does C'.

Step 2: In the next step we form a fully recursive half- plane two port $T^{(2)}$ by interchanging the two output terminals in each port of T' as shown in figure 4.3. It can be easily shown that $T^{(2)}$ can then be written in terms of pseudopolynomials $A^{(2)}$, $B^{(2)}$, $C^{(2)}$ and the rational function $D^{(2)}$ in standard form as expressed in property 3.1′, where

$$A^{(2)} = A', \quad B^{(2)} = C', \quad C^{(2)} = B', \quad D^{(2)} = -D' \qquad (4.8)$$

Step 3: A GGM section is then extracted from the two-port with chain matrix $T^{(2)}$ by iterating step 1 on $T^{(2)}$ to get a fully recursive half-plane lossless two-port chain matrix

$T^{(3)}$. If pseudopolynomials $A^{(3)}$, $B^{(3)}$, $C^{(3)}$ and rational function $D^{(3)}$ represent $T^{(3)}$ in standard form as in property 3.1′ then $B^{(3)}$ would have a factor z_2. Also, since from (4.4b) and (4.8) we have $C^{(3)} = C^{(2)}k_2^{(3)} = B'k_2^{(3)}$, where $k_2^{(3)}$ defines the GGM section presently extracted, and B' has a factor z_2 we conclude that $C^{(3)}$ has a factor z_2. From this and the fact that $A^{(3)}\tilde{A}^{(3)} = B^{(3)}\tilde{B}^{(3)} + C^{(3)}\tilde{C}^{(3)}$ it follows that $A^{(3)}\bar{A}^{(3)} = 0$ for $z_2 = 0$ and for arbitrary z_1. Since $\bar{A}^{(3)} \neq 0$ for $|z_1| = 1$ and $|z_2| < 1$, we conclude $A^{(3)} = 0$ for $z_2 = 0$ and for arbitrary z_1. Consequently, $A^{(3)}$ has a factor z_2 and it is possible to write, for some pseudopolynomials $A^{(4)}$, $B^{(4)}$ and $C^{(4)}$ that

$$A^{(4)} = z_2A^{(3)},\ B^{(4)} = z_2B^{(3)},\ C^{(4)} = z_2C^{(3)} \tag{4.9}$$

Step 4: The last step in the synthesis cycle is to extract a z_2 type delay from $T^{(3)}$ as in figure 4.4 to produce a two-port with chain matrix $T^{(4)}$, which in standard form can be expressed in terms of $A^{(4)}$, $B^{(4)}$, $C^{(4)}$. Furthermore, $\bar{A}^{(4)} = \bar{A}^{(3)} \neq 0$ for $|z_1| = 1$, $|z_2| < 1$; $\bar{A}^{(4)} = \bar{A}^{(3)}$ can have at most finitely many zeros on $|z_1| = |z_2| = 1$. Also, it follows from (4.9) and losslessness of $T^{(3)}$ that

$$\tilde{A}^{(4)}A^{(4)} = \tilde{B}^{(4)}B^{(4)} + \tilde{C}^{(4)}C^{(4)},\quad \deg_2 B^{(4)} \leq \deg_2 A^{(4)},\ \deg_2 C^{(4)} \leq \deg_2 A^{(4)} \tag{4.10a,b}$$

Thus, the two-port associated with $T^{(4)}$ is fully recursive half-plane lossless. Furthermore, note that

$$\deg_2 A^{(4)} = \deg_2 A^{(3)}-1 = \deg_2 A^{(2)}-1 = \deg_2 A'-1 = \deg_2 A-1,$$

where the first equality follows from (4.9); the second and the fourth from the fact that in step 1 we have $n_A = n_{A'}$; and the third from (4.8).

Consequently, after iterating $\deg_2 A$ times the cyclic algoritm described in Steps 1 through 4, we obtain a lossless chain matrix T_f independent of z_2, which in standard form is described by $A_f = A_f(z_1)$, $B_f = B_f(z_1)$, $C_f = C_f(z_1)$ and $D = D_f(z_1)$.

Terminal Step: In the final step we extract another GGM section as in Step 1 to produce a fully recursive half-plane lossless two-port S_0 with $(S_0)_{11}=(S_{22})=0$, where $(S_0)_{12}$, $(S_0)_{21}$ are both 1-D all-pass functions (not necessarily stable). The realization for such a two-port is shown in figure 4.5. The composite filter structure is shown in figure 4.6.

5. DESIGN OF 2-D FULLY RECURSIVE HALF-PLANE FILTERS

Our design proceeds by requiring the transfer function $S_{21} = C/\bar{A}$ of the lossless two-port S to have the desired characteristics. However, unlike the corresponding problem in 1-D, due to nonfactorability of m-D polynomials it is in general not possible to find a pseudopolynomial B satisfying (3.5a) from A and C. To circumvent this problem it will be further assumed that the two-port is either symmetric i.e., $S_{11} = S_{22}$, $S_{21} = S_{22}$ or antimetric i.e., $S_{11} = - S_{22}$, $S_{21} = S_{12}$. We next define two rational functions S_1 and S_2 as in (5.1) and (5.2) respectively for symmetric or antimetric two ports.

$$S_1 = (B + C)/\bar{A},\ S_2 = (B - C)/\bar{A};\ S_1 = (B + jC)/\bar{A},\ S_2 = (B- jC)/\bar{A} \tag{5.1a,b; 5.1′a,b}$$

It can then be shown that S_1 and S_2 in (5.1) must be a fully recursive half-plane all-pass function and satisfy properties analogous to those in Property 2.3. Note that even if A, B, C are real rational functions, S_1 and S_2 are real in (5.1ab) but not in (5.1′a,b). Thus, a symmetric filter can be realized by making use of the relation $S_{21} = C/\bar{A} = (S_1-S_2)/2$, where the one-ports S_1 and S_2 can be realized as one ports via algorithms similar to that of section 4. Although $S_{21} \equiv C/\bar{A} = -j(S_1-S_2)$ holds true in the antimetric case, a realization in terms of this last mentioned equation is not feasible unless complex filter realizations are called for. In this case, the pseudopolynomials A, B, C which are real, can be found from (5.1'a,b) and subsequently S_{21} can be realized as being embedded in a real two-port S described by A, B, C in standard form. The design problem then boils down to appropriately choosing the real 1-D rational functions D_1, D_2, and real pseudopolynomials A_1, A_2 so that the frequency response requirements on $|S_{21}|$ are satisfied. This latter step may be carried out by using numerical optimization (e.g., Levenberg-Marquadt). Note that if $|S_{21}(\omega_1,\omega_2)|$ possesses either the fan-type or the circular-type symmetry then $|S_{21}(\omega_1,\omega_2)|$ must have, in particular, the so called quadrantal symmetry i.e., $|S_{21}(\omega_1,\omega_2)|$ is unaltered if the signs of either ω_1 or ω_2 or both are changed. This requirement coupled with the stability property of S_{21} demands that the denominater pseudopolynomial of S_{21}, and thus A_1 and A_2 in (5.3) satisfy certain factorability property. It then proves to be expedient to carry out the optimization procedure on these factors from the point of view of the number of parameters to be optimized.

REFERENCES

[1] J. Lee and J.W. Woods, Design and implementation of two dimensional fully recursive digital filters, IEEE Trans. on ASSP, Vol. 34, No. 1, pp. 178-191, Feb. 1986.

[2] A. Fettweis, Wave digital filters: Theory and practice, Proc. IEEE, vol. 74, pp. 270-327, Feb. 1986.

[3] E. F. Deprettere and P. Dewilde, Orthogonal cascade realization of multiport digital filters, Int. J. Circuit, Th. and Appl., vol. 8, pp. 254-272, July 1980.

[4] A. Fettweis, Multidimensional digital filters with closed loss behavior designed by complex network theory approach, IEEE Trans. on CAS, vol. 34, No. 4, pp. 338-345, April 1987.

[5] H. Lev-Ari, S. R. Parker, T. Kailath, Parametrization of 2-D filters, Int. Symp. on CAS, pp. 1022-1025, Philadelphia, 1987.

[6] N. K. Bose, Applied Multidimensional Systems, Van Nostrand Reinholdt, 1982.

[7] S. Basu, A. Fettweis, On the factorization of transfer scattering matrices of lossless two-ports, IEEE trans. on CAS, vol. 32, pp. 925-934, Sept. 1985.

[8] S. K. Rao and T. Kailath, Orthogonal digital filters for VLSI implementation, IEEE Trans. on Circuits and Systems, Vol. 31, pp. 933-945, Nov. 1984.

[9] A. Fettweis, Network theory approach to designing lossless (orthogonal) digital filter structures, IEEE Int. Symposium on CAS, pp. 154-156, Philadelphia, 1987.

[10] R. J. Walker, Algebraic Curves, Springer Verlag, New York, 1968.

[11] S. Basu and A. Fettweis, New results on stable multidimensional polynomials, Part II: Discrete case, to appear in IEEE trans. on Circuits and Systems, Nov. 1987.

[12] V. Belevitch, Classical Network Theory, Holden-Day, San Francisco, 1968.

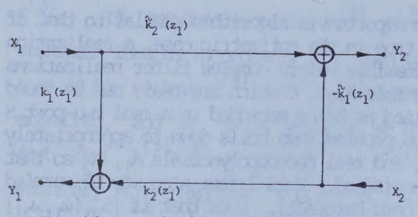

X_1
$\tilde{k}_2(z_1)$
Y_2
$k_1(z_1)$
$-\tilde{k}_1(z_1)$
Y_1
$k_2(z_1)$
X_2

Figure 4.1: A generalized Gray Marked section (G G M)

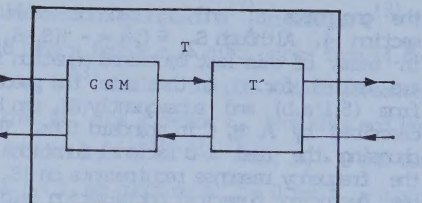


Figure 4.2: Step 1; G G M extraction

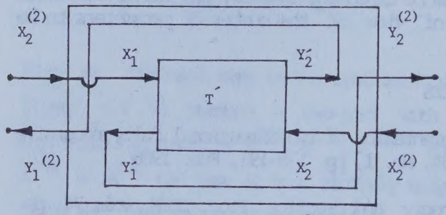

$X_2^{(2)}$
X_1'
Y_2'
$Y_2^{(2)}$
T´
$Y_1^{(2)}$
Y_1'
X_2'
$X_2^{(2)}$

Figure 4.3: Step 2; Two-port $T^{(2)}$ obtained from T´

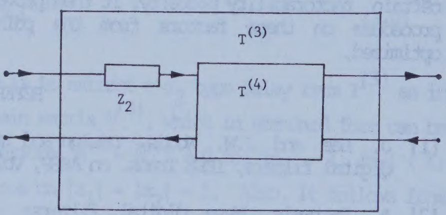


Figure 4.4: Step 4; Z_2 - delay extraction

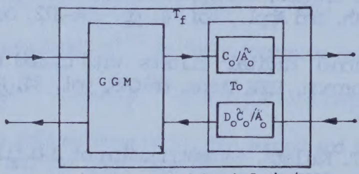

T_f
G G M
$C_0/\tilde{A}_0$
To
$D_0\tilde{C}_0/\tilde{A}_0$

Figure 4.5: Terminal Step in Synthesis

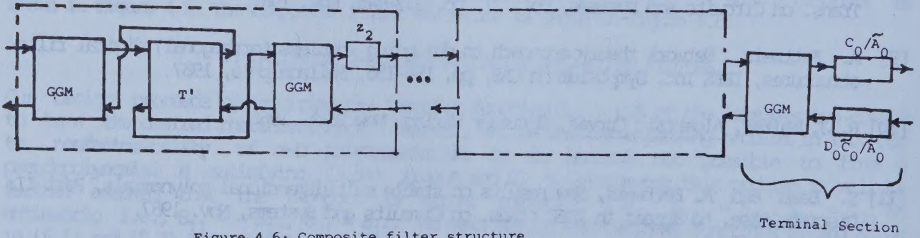

GGM
T'
GGM
z_2
•••
GGM
$C_0/\tilde{A}_0$
$D_0\tilde{C}_0/\tilde{A}_0$
Terminal Section

Figure 4.6: Composite filter structure

CONCURRENT PROCESSING WITH RESULT SHARING: AN EFFICIENT COMPUTING PARADIGM

S. KRISHNAPRASAD
Computer Science & Engg., Auburn University
Auburn, Alabama 36849

ABSTRACT

A major aspect of efficient problem solving is to avoid redundant recomputations. This paper identifies the need for and ways to incorporate both problem structure and problem dynamics, in the context of concurrent processing, for fast and efficient problem solving. The notion of horizontal locality and vertical locality are introduced to capture the essence of problem dynamics. An efficient computing paradigm called Concurrent Processing with Result Sharing is suggested to represent concurrency and result sharing in general problem solving.

1. MOTIVATION

In an era of inexpensive, high-speed semiconductor technology and with the advent of VLSI techniques, a whole range of large scale problem solving is becoming a practical reality. The need to solve problems faster has been partially met by the advent of multiprocessors, parallel processors, and distributed processors [Hwa 84]. These schemes try to reduce the raw computation time of the problem by exploiting the inherent concurrency existing in the problem structure. Such a problem solving strategy corresponds to spatial cooperation wherein a set of processing elements cooperate in space towards the problem solving goal.

Another dimension in problem solving is pipelining wherein resource sharing is exploited to achieve increased throughput. Pipelining exploits the property of a stream of problem instances demanding similar resources in an orderly way thus achieving a high resource utilization and throughput.

Yet another dimension in problem solving is virtual computation [Kri 82a, 82b] which represents an explicit sharing of past results (that is, outcome of past computational activities). It essentially corresponds to table look-up (partial or full) type of strategy. We say an instance of computation virtual if no actual or direct computation is performed but table look-up is used. We shall refer to this as temporal cooperation as past activities are usefully exploited. In virtual computation or past result sharing the notion of result as a resource is fundamental. The idea here is to avoid duplicate or redundant computations. The major advantages of virtual computation are:

*higher throughput by avoiding expensive recomputations.

*by avoiding redundant activities we are correspondingly reducing the demand on computational and communication resources which could be used for other computations thereby further improving the throughput. Thus in a concurrent processing environment with past result sharing there is a large scope for efficient processing.

*some problem instances will be solved faster than their isolated lower bound time. This is not an illusion but a direct consequence of temporal cooperation.

We broadly identify two orthogonal abstractions which are fundamental for fast and efficient problem solving:

*concurrent processing which includes parallel processing, multiprocessing, distributed processing, and pipelining.

*result sharing which represents saving and sharing of past computed results. Cache memory techniques and expert systems represent two extremes of saving and sharing of past activities.

Knowledge of problem structure is essential not only to solve the problem using concurrent processing but also to characterize the intra-instance sharing aspects of it. Partial evaluation, common subexpressions, dynamic programming paradigm, etc., are all techniques which exploit problem structure. A functional network model like a data flow graph nicely captures the problem structure abstraction. The effectiveness of decomposition schemes for concurrent processing fails once we take into account the problem dynamics. Not all problem variables or parameters change at every instance of problem evaluation and further, variables may change at different rates. Thus activities and change pattern of variables have direct impact on recomputation effort necessary for successive instances of problem request. It is our claim that for fast and efficient computation we need to combine both concurrent processing and result sharing leading to a unified model called Concurrent Processing with Result Sharing (CPRS) model. Section 2 considers the decomposition aspects for concurrent evaluation with and without problem dynamics. Section 3 introduces the CPRS model of computation. Conclusions are included in Section 4.

2. DECOMPOSITION WITH AND WITHOUT DYNAMICS

The main motivation for concurrent processing is fast evaluation of the problem. To our knowledge, saving and sharing of past computational results in the context of concurrent processing, on a dynamic basis, have not been considered by researchers. The major motivation for sharing past results is to exploit problem dynamics. Traditional decomposition schemes for concurrent processing (CP decomposition for short) use problem semantics to transform the problem structure. When the same problem is to be recomputed repeatedly, with different variable values, CP decompositions may become both resource intensive and slower compared to the alternative decomposition and model of evaluation we are proposing. The decomposition which takes into account both problem structure and problem dynamics is called CPRS decomposition and the model of computation is called CPRS model. Problem dynamics are partly abstracted in terms of activities or velocities of variables upon which given problem depends.

2.1 CP Decomposition

Decomposition schemes suitable for concurrent evaluation of expressions (boolean, arithmetic, matrix) have been extensively studied by many reseachers ([Kar 63], [Cur 63], [Hel 66], [Sto 67], [Ram 69], [Kuc 72], [Ram 73], [Mur 73], [Bae 73], [Ris 82], [Bra 82a, 82b]). Problem structure is exploited to realize faster evaluation and intra-instance result sharing. The latter uses the notion of common subexpressions and factorization to share substructures in the overall evaluation net. These substructures could represent hardware or software modules.

Decomposing an arithmetic expression into a CP net is necessary to evaluate the expression on a multioperation machine. The general approach is to represent the parse tree of the expression and use algebraic properties like commutativity, associativity, and distributivity to transform this structure with the goal of reducing the height of the tree. These techniques can be classified under tree height reduction algorithms, to borrow the term from [Kuc 78]. Ramamoorthy et al [Ram 73] have considered several compilation techniques for recognition of parallel processable tasks in arithmetic expressions. One of these transforms the Polish string representation of the given expression from which a maximal parallel form is derived. Properties of commutativity and associativity (but not distributivity) are fully exploited during the transformation. Optimal algorithms are given to generate minimum level tree and minimum execution time tree from the maximal parallel form of the input expression. CP decomposition of a restricted class of matrix expressions is reported in [Mur 73].

2.2 CPRS Decomposition

Though problem structure plays a key role in determining problem solving strategy, various properties of the individual variables involved could be exploited to alter the problem solving structure (decomposition net) and strategy with the goal of efficient processing. Some of these properties are: the cost of evaluating or determining variable values [Bre 75a, 75b], the degree of dependence of the function on the variable [Mor 80], the activity of the variable which captures the notion of how frequently the variable changes its value in the problem request stream [Kri 87]. The traditional CP decomposition totally ignore problem dynamics and hence potential past result sharing. Problem dynamics abstracted as activities of variables can be exploited to reduce both average computational time and average space resources if only the decomposition scheme uses variable dynamics to alter the evaluation net. This corresponds to the CPRS decomposition net as identified in Section 2. Of course, to exploit dynamics we need to save past computational results. The most recently computed results at each node is assumed to be saved in the CPRS net.

Consider the example given in [Kuc 72]. To evaluate the expression E=A*(B*C*D+E) the CP decomposition given in Figure 1 is optimal in terms of tree height (note distributivity has helped CP decomposition). But suppose during repeated evaluation of E if variable A alone changes most frequently and other variables are relatively slow in changing, then intuitively it is clear that the decomposition given in Figure 2 is much better, both in terms of space and time requirements, compared to the net of Figure 1. Thus, in general, CPRS decomposition scheme considers both problem structure properties and problem dynamics, and generates an optimal or near optimal decomposition with respect to some chosen measures like average weighted tree height of CPRS net and average number of nodes to be recomputed. Such algorithms are discussed elsewhere [Kri 87] for arithmetic-type expression evaluation wherein variable dynamics are abstracted as activity values and algebraic properties of commutativity, associativity, and distributivity are exploited.

3. CPRS MODEL OF COMPUTATION

In the context of spatial cooperation, the terms vertical concurrency and horizontal concurrency are used to describe, respectively, concurrent processing and pipelining. In temporal cooperation we introduce two dimensions: horizontal locality (HL) and vertical locality (VL). HL and VL, which are discussed below, are two key aspects of problem dynamics represented in the CPRS model which are useful for result sharing.

3.1 Horizontal Locality

As the computation of a problem evolves, the successive instances of the problem in the problem request stream, over a period of time, will in general display locality of values, that is, variable values will be repeated. We refer to this aspect of problem dynamics by the term horizontal locality and the resultant sharing aspect as HL-type sharing (Figure 3). Exploiting HL implies saving previously computed results in a local result database (RDB), at each node, which stores [instance descriptor, result] pairs. When a computation instance f(X) is requested at time t we first look into the RDB. If we succeed then there is no need for scheduling this request as it is solved virtually. Only if the look up fails do we need to perform direct computation. With a high hit ratio into the RDB the effectiveness of HL-type sharing will be large. For practical reasons the size of the RDB will be limited. The successful use of cache memories illustrates classic example of HL-type sharing (the 'computation' involved here is memory access).

3.2 Vertical Locality

Another aspect of problem dynamics is the fact that between two successive problem instance requests very few variables, in general, are changed. This implies that we have to recompute only a partial graph of the overall computation graph, determined by the variables that have changed and their influence on the evaluation net. We refer to this aspect of problem dynamics by the term vertical locality and the resultant sharing aspect as VL-type sharing (Figure 4). To support VL-type sharing we need to save the most recently computed result at each node of the evaluation net. As this represents the result of a single instance of computation VL can be exploited easily and inexpensively.

In Figure 4 an evaluation net for an arbitrary computation f(x1,x2,...,x7) is given where the dots inside each node represent the most recently computed result at that node. For simplicity, let the space and time requirements of each node be one unit each. If no sharing is employed, then each instance of computation takes 5 units of space and 3 units of concurrent time. Suppose between the previous instance and current instance only variables x1 and x2 have changed. Then with VL-type sharing we need to recompute only the nodes 1 and 5 and the computation needs only 2 units of space and 2 units of time. Thus the current instance is solved both faster and with less resources and we claim that this truly represents the spirit of efficient computation. The only other support we need is to detect the variables that have changed and schedule only those nodes that are in the recomputation graph.

3.3 CPRS Model

The most natural way to model concurrent evaluation is the data flow model [Com 82], [Tre 82]. The asynchrony and functional aspects of the data flow model allow us to exploit all the concurrency exhibited in the data flow graph of the computation. The CPRS model is an extension of the data flow model with the abstractions for exploiting problem dynamics. A computation is represented by a CPRS graph or net (Figure 5). Each node of the graph has a local RDB supporting HL and most recently computed result (dot) supporting VL. Furthermore, the structure of the net will display full concurrency after taking into consideration the transformations on the net due to problem dynamics consideration. The decomposition algorithm, if one possible for a given problem, uses the dynamics information to push the fastest changing variables as close to the root of the evaluation net as possible. It is easy to show that the structure of the best CPRS net may correspond to the structure of the worst CP net with the former mode of evaluation better than the latter. Several measures could be defined for the CPRS graph. The weighted length of recomputation path, the weighted size of the recomputation graph, etc.

In addition to the support needed to execute the data flow model of computation the CPRS model requires fast capabilities to detect the recomputation graph (based on VL) and schedule only the necessary computations. In addition fast associative RDB support (for HL) may be useful in some cases. These aspects of the CPRS model are highly application dependent.

An interesting feature of CPRS mode of computation is that the time needed to compute and the space needed in terms of the computational resources are both relative quantities. The relativity of space and time requirements is a direct consequence of temporal cooperation exploiting the problem dynamics. This implies rethinking to be done on the traditional notion of complexity of computation.

4. CONCLUSIONS

This paper has suggested motivation for incorporating past result sharing for exploiting the problem dynamics in the context of concurrent processing. HL and VL are two key features of dynamics which are part of the CPRS model of computation. Decomposition of a problem based on variable dynamics is a difficult problem in general and more work is needed in this regard. Of course, the effectiveness our model need to be demonstrated for non-trivial real life applications. We strongly feel that while concurrent processing is widely recognized as key to fast problem solving one should not ignore the potential for past result sharing based on problem dynamics. We strongly encourage researchers to identify specific real life applications where CPRS model of computing is both faster and efficient compared to CP model.

REFERENCES

[Bae 73] "A survey of some theoretical aspects of multiprocessing," J.L.Baer, Computing Surveys, Mar 1973, pp 36-48.

[Boz 75] "Certain properties of boolean differentials and the activities of the arguments of boolean functions ...," Bozoyan, Engg. Cybernetics. 13, 5, Sep 1975, pp 108-120.

[Bra 82a] "Fast recursive boolean function manipulation," Brayton et al., IEEE Int. Conf. on Circuits & Systems, 1982, pp 58-62.

[Bra 82b] "The decomposition and factorization of boolean expressions," Brayton et al., ibid, pp 49-54.

[Bre 75a] "A branch-and-bound algorithm to obtain an optimal evaluation tree for monotonic boolean functions," Breitbart et al., ACTA Informatica, Springer-Verlag, 1975, pp 311-319.

[Bre 75b] "Algorithms for fast evaluation of boolean functions," Breitbart et al., ibid, pp 107-116.

[Com 82] Special issue on data flow systems, IEEE COMPUTER, Feb 1982.

[Cur 63] "Generalized tree circuits: the basic building block of an extended decomposition theory," H.A.Curtis, Jour. of ACM, 1963, pp 562-581.

[Hel 66] "Parallel processing of algebraic expressions," H.Hellerman, IEEE Trans. Electronic Computers, Feb 1966, pp 82-91.

[Hwa 84] Computer Architecture and Parallel Processing, K.Hwang and F.A.Briggs, McGraw-Hill Inc., 1984.

[Kar 63] "Functional decomposition and switching circuit design," R.M.Karp, SIAM Jour., Jun 1963, pp 291-335.

[Kri 82a] "Problem solving through result sharing: a unified notion of virtual computation," S.Krishnaprasad, Internal communication, Dept. of Computer Sci., SMU, Dallas, Spring 1982.

[Kri 82b] "Virtual computation: an update on result sharing schemes," S.Krishnaprasad, Internal communication, Dept. of Computer Sci., SMU, Dallas, Fall 1982.

[Kri 87] "Concurrent processing with result sharing: a unified view of efficient computations," S.Krishnaprasad, Doctoral research in progress, Dept. of Computer Sci., SMU, Dallas.

[Kuc 72] "On the number of operations simultaneously executable in FORTRAN-like programs and their resulting speedup," Kuck et al., IEEE Trans. Computers, Dec 1972, pp 1293-1310.

[Kuc 78] The Structure of Computers and Computations, Vol I, D.J.Kuck, John Wiley & Sons, 1978.

[Mor 80] "The activity of a variable and its relation to decision trees," Moret et al., ACM TOPLAS, Oct 1980, pp 580-595.

[Mur 73] "On the time required for a sequence of matrix products," Y.Muraoka & D.J.Juck, Comm. ACM, Jan 1973, pp 22-26.

[Ram 69] "A survey of techniques for recognizing parallel processable streams in computer programs," C.V.Ramamoorthy & M.J.Gonzalez, FJCC, 1969, pp 1-15.

[Ram 73] "Compilation techniques for recognition of parallel processable tasks in arithmetic expressions," Ramamoorthy et al., IEEE Trans. Computers, Nov 1973, pp 986-998.

[Ris 82] "Staggered input networks: an approach to automatic logic decomposition," R.H.Risch, IEEE Conf. on Circuits & Systems, 1982, pp 55-57.

[Sto 67] "One-pass compilation of arithmetic expressions for a parallel processor," H.S.Stone, Comm. ACM, Apr 1967, pp 220-223.

[Tre 82] "Data-driven and demand-driven computer architecture," Treleaven et al., ACM Computing Surveys, Mar 1982, pp 93-143.

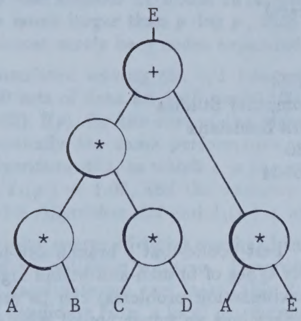


Figure 1: CP net for E

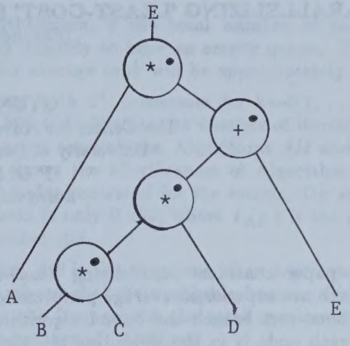


Figure 2: CPRS net for E

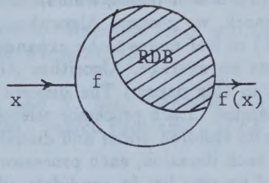


Figure 3: HL-type sharing

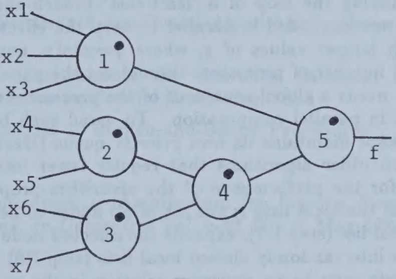

x1
x2
x3
1
x4
2
x5
x6
3
x7
4
5
f

Figure 4: VL-type sharing

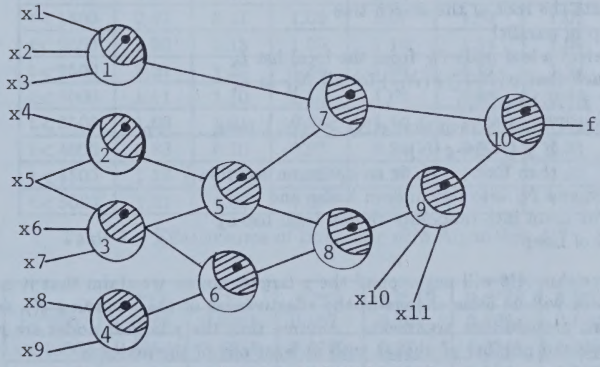

x1
x2
x3
1
x4
2
x5
x6
3
x7
x8
4
x9
5
6
7
8
9
10
f
x10
x11

Figure 5: CPRS graph

PARALLELIZING 'LEAST-COST' BRANCH-AND-BOUND ALGORITHMS
(Extended Abstract)

GYUNGHO LEE
The Center for Advanced Computer Studies
University of Southwestern Louisiana
P.O. Box 44330
Lafayette, LA. 70504

This paper considers parallelizing 'least-cost' (or 'best-bound-first') branch-and-bound algorithms, which are superior in average performance to other types of branch-and-bound algorithms [3]. A typical least-cost branch-and-bound algorithm (for maximization problems) can be stated as follows: For each node N in the serch tree the algorithm determines an upper bound g(N) and a lower bound f(N) for all possible costs represented by the nodes that can be spawned from the node N, and proceeds the serach by selecting the node with the largest value of g.

In making the loop of a 'least-cost' branch-and-bound algorithm parallel, several nodes with least costs are expanded in parallel to keep the effectiveness of the algorithm [1][2][4]: Expand the q nodes with largest values of g, where q=min(p, number of leaf nodes in the search tree) (p will denote the number of processors throughout the paper). This approach, which we will refer as Algorithm *Al1*, needs a global consensus of the processors to find out best nodes, which can be a potential bottleneck in parallel computation. To avoid such bottleneck, we propose Algorithm *Al2*, in which each processor maintains its own priority queue (*local list*) of leaf nodes to be expanded. (Since we know of no other algorithms that require fewer iterations on average, Algorithm *Al1* serves as a yardstick for the performance of the algorithm proposed in this paper.) The disjoint union of the nodes in all the local lists is the set of all nodes to be expanded. Each processor selects a node from its own local list (step R3), expands the selected node into its children nodes and distributes the children nodes into randomly chosen local lists (step R6). At each iteration, each processor checks if the selected node may be an optimum solution in the range of its own list (a *candidate* of an optimum solution) (step R4). When a processor finds a candidate of an optimum solution, we need a global concensus to check if the candidate is a real optimum solution (step R5).

Algorithm Al2

```
(R1)  Create the root of the search tree
(R2)  Loop in parallel
(R3)     Select a leaf node $N_i$ from the local list $L_i$
         such that $g(N_i) \geq g(N_{ij})$ for all $N_{ij}$ in $L_i$
(R4)     If $f(N_i) = g(N_i)$
(R5)     then { Find N such that $g(N) \geq g(N_i)$, i=1, . . . , p
                If $f(N_i) = g(N)$
                then Exit - - $N_i$ is an optimum solution }
(R6)     Expand $N_i$ into its children nodes and
         Put them into randomly chosen local list $L_k$
      End of Loop
```

Although, Algorithm *Al2* will not expand the p largest nodes, we claim that it is still reasonable in the sense that there will be little change in the effectiveness of the bounding g(), which guides the search. We have two probabilistic arguments. Assume that the p largest nodes are randomly distributed. Then the expected number of queues with at least one of the nodes is

$$(1 - (1 - 1/p)^p) \cdot p \approx 0.63 \cdot p .$$

This shows that a significant fraction of the largest p nodes will be expanded at each iteration. Another way to see this is to consider the average rank of a node being expanded. In the ideal situation, the p largest nodes will be expanded and their average rank is $(p+1)/2$. For the probabilistic scheme, the average rank of the largest node in a queue is

$$\sum^{n} i \cdot Prob \text{ (rank is in the queue and no rank is smaller than i)}$$
$$\leq \sum^{\infty} i \cdot \frac{1}{p}\left(1 - \frac{1}{p}\right)^{i-1} = p ,$$

where n is the number of nodes in a queue. Furthermore, if the total number of nodes to be expanded is much larger than $p \cdot \log p$, then it is very unlikely to have an empty queue. This means there will almost surely be p nodes expanded, and their average rank will be approximately p.

We simulated solving the 0/1 knapsack problem with 2^k processors for k=0,1, . . . , 6. We generated 30 sets of data for each n=50,100,150,200,300,400. We use the number of iterations of the loop (step R2), I(p), for the cost of the algorithm. In our simulations, Algorithms *Al1* and *Al2* have exhibited virtually the same performance. Table 1 shows the effectiveness of Algorithm *Al2* compared to Algorithm *Al* 1, in which t is the number of nodes generated for the search. On average the ratio $I_2(p)/I_1(p) = 1.03$, and the variance of the ratio is only 0.053, where $I_2(p)$ is the number of iterations with Algorithm *Al2* and $I_1(p)$ is with Algorithm *Al1*.

The major source of serial overhead in Algorithm *Al2* is the occasions when a processor finds a candidate of an optimum solution. However, the overhead shall be negligible because the occasions of finding a candidate are rare (see Table 2, in which the frequency is the total number of occasions of finding a candidate from a local list divided by the number of iterations).

REFERENCES

[1] T. Lai and S. Sahni, "Anomalies in Parallel Branch-and-Bound Algorithms", *Comm. of ACM*, vol.27, pp.594-602, 1984

[2] J. Mohan, "A Case Study in Parallel Computation: the Traveling Salesman Problem", *Proceeding of the 1983 International Conference on Parallel Processing*, 1983

[3] D. R. Smith, "Random Trees and the Analysis of Branch-and-Bound Procedures", *Journal of ACM*, vol. 31, no. 1, 1984

[4] B. Wah and Y. Ma, "MANIP - A Multicomputer Architecture for Solving Combinatorial Extremum - Search Problems", *IEEE Trans. on Computers* Vol. c-33, No. 5, May 1984

$I_2(p)/I_1(p)$						
	$p=2$	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
t<500	1.08	1.06	1.06	1.04	1.02	1.02
t<1000	1.03	1.13	1.10	1.04	1.01	1.01
t<1500	0.87	0.91	1.09	1.07	1.04	1.02
t<2000	1.00	1.13	1.09	1.10	1.14	1.06
t<2500	1.21	1.27	1.39	1.30	1.27	1.24
t<3000	1.11	1.10	1.10	1.02	1.08	1.03
t<3500	1.03	1.01	0.88	1.10	1.01	0.94
t<4000	0.88	0.70	0.87	0.84	0.87	0.81
t<4500	1.33	1.11	1.07	0.90	1.06	1.03
t<5000	1.01	1.05	1.06	1.06	1.07	1.04

Table 1. Effectiveness of Bounding with Algorithm *Al2*

	$p=2$	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
t<500	0.0153	0.0385	0.102	0.102	0.102	0.102
t<1000	0.0054	0.0259	0.0722	0.0745	0.0745	0.0851
t<1500	0.0028	0.0327	0.0935	0.1628	0.189	0.2126
t<2000	0.0022	0.0083	0.0241	0.1037	0.1443	0.1951
t<2500	0.0016	0.0637	0.0945	0.2836	0.392	0.4432
t<3000	0.0044	0.0086	0.0144	0.0241	0.0530	0.0692
t<3500	0.0013	0.0046	0.0033	0.0109	0.0227	0.0747
t<4000	0.0012	0.0078	0.1102	0.1854	0.2303	0.2206
t<4500	0.0025	0.0098	0.0286	0.1111	0.129	0.18
t<5000	0.0017	0.0017	0.0027	0.004	0.0053	0.0061

Table 2. Frequency of Finding a Candidate of the Solution

THE SIMPLE SIMON PROGRAMMING ENVIRONMENT:
A Status Report[1]

JANICE E. CUNY　DUANE A. BAILEY
JOHN W. HAGERMAN　ALFRED A. HOUGH

Computer and Information Science Department,
University of Massachusetts, Amherst, MA

Abstract: The Simple Simon Programming Environment is designed to serve as a testbed for the development of programming support specific to highly parallel computation. It consists of a set of programming tools and a kernel. The kernel contains a central database to facilitate the prototyping of new tools. We describe three such tools here: a graph editor for describing process interconnection structures; a graph assistant for creating aesthetic graph layouts; and a "pattern-oriented" parallel debugger.

1. Introduction

The Simple Simon Programming Environment is designed to serve as a testbed for the development of programming support specific to highly parallel computation. Such support might include mechanisms for the specification of multiple processes, the control of interprocess communication and synchronization, the mapping of logical structures onto physical interconnections, and the debugging of closely coupled processes for both correctness and performance.

Since the *process of programming* itself changes as more sophisticated tools and methodologies become available, high-level programming environments must evolve with use [1,2]. We have begun with a rudimentary – but extensible – parallel environment, called Simple Simon that runs as the front end to the Simon Multiprocessor Simulator[3,4]. Simon is an event-driven simulator of concurrent, cooperating processes that are statically specified and individually programmed. It can model a variety of communication disciplines, executing a parallel application program as if each constituent process was running on its own processor. The Simple Simon environment makes Simon accessible to a local community of students and researchers. In turn, these programmers form the target community for the design and evaluation of successively higher-level extensions, making Simple Simon the testbed for our research.

Simple Simon consists of a set of programming tools together with a kernel that supports the prototyping of new tools as shown in Figure 1. The kernel provides the core facilities of the environment. The tools provide optional features of the environment that can be replaced or modified at the user's discretion. We discus prototypes of three tools: a graphical database editor (GDE) that provides facilities for describing programs graphically; a graph assistant (GA) that uses heuristics to assist the programmer in designing aesthetic displays of interconnection structures; and a "pattern-oriented" debugger (Belvedere) that provides animations of system behavior. They are described in Sections 2, 3, and 4 respectively. We summarize the contributions of Simple Simon in Section 5.

---

[1] This work was supported by the Office of Naval Research under contract N000014-84-K-0647.

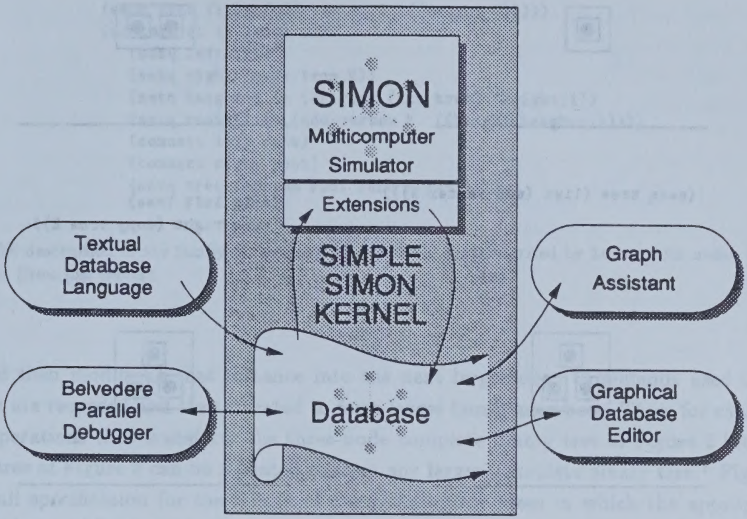


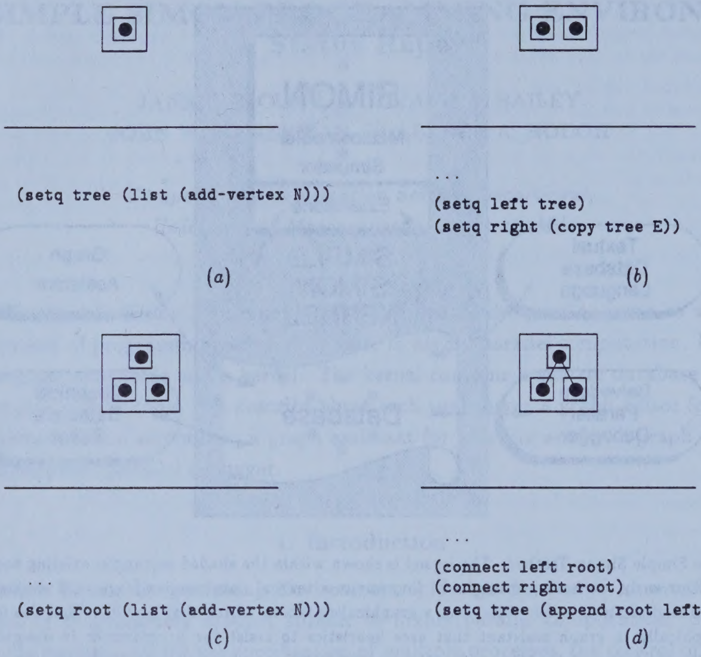
Figure 1: The Simple Simon Testbed. The kernel is shown within the shaded rectangle; existing tools are shown within ovals. Currently, we have prototypes of four tools: a textual database language that enables the user to enter information directly into our database; a graphical database editor that provides facilities for describing programs graphically; a graph assistant that uses heuristics to assist the programmer in designing aesthetic displays of interconnection structures; and a "pattern-oriented" parallel debugger that provides animations of system behavior. Arrows in this diagram indicate the transfer of data.

2. GDE: The Graph Editor

We expect that the Graphical Database Editor will be used to input most graphical structures but our initial version is limited to the description of process interconnection structures as labeled graphs in which the nodes represent processes and the edges represent logical communication channels.

Despite the fact that most existing parallel algorithms use one of a very few interconnection structures, we believe that the explicit description of interconnections is important. Such descriptions lend themselves to informative graphical displays, suggest information useful in the mapping from logical to physical structures [5], and provide redundancy for error analysis (making it possible, for example, to check for mismatched ports, dangling channels, path redundancy, or disconnected components). Explicit specifications enable the programmer to create the nonstandard graphs that are often needed when existing algorithms are composed into actual programs. In addition, they provide convenient mechanisms for attaching labels (for example, processor id, code names, or parameters) to graph components.

Because parallel programs are often designed in-the-small and then scaled, programmers need facilities for describing not just graphs, but *graph families.* GDE is an early prototype of such a facility; it has been implemented so as to experimentally determine the types of graph operations that are most useful. Using GDE, a graph family is described by drawing its smallest



```
(setq tree (list (add-vertex N)))
```

(*a*)

```
...
(setq left tree)
(setq right (copy tree E))
```

(*b*)

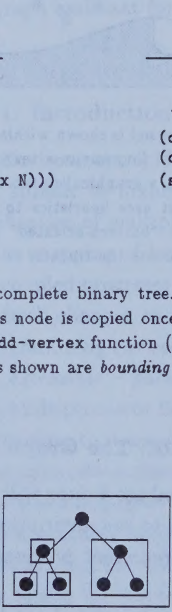
```
...
(setq root (list (add-vertex N)))
```

(*c*)

```
...
(connect left root)
(connect right root)
(setq tree (append root left right))
```

(*d*)

Figure 2: GDE description of a three node complete binary tree. The description starts with a single node created with the **add-vertex** function (*a*); this node is copied once to create a pair of siblings using the **copy** function (*b*); a root node is added using the **add-vertex** function (*c*); and the root is connected to its children using the **connect** function (*d*). The rectangles shown are *bounding boxes* used for spatial reference.



```
...
(setq left tree)
(setq right (copy tree E))
(setq root (list (add-vertex N)))
(connect left root)
(connect right root)
(setq tree (append root left right))
```

Figure 3: GDE description of a seven node complete binary tree constructed from the three node tree described above. Note that these instructions could be iterated to form successive members of the family of complete binary trees.

```
(setq tree (list (add-vertex N '((height 0)))))
(dotimes (i (- level 1))
   (setq left tree)
   (setq right (copy tree E))
   (setq height+1 (+ (extract (car tree) 'height)1))
   (setq root (list (add-vertex N '((height height+1)))))
   (connect left root)
   (connect right root)
   (setq tree (append root left right)) )
```

Figure 4: GDE description of the family of complete binary trees, parameterized by `level`, with nodes labeled by their height (from the leaves).

instance and then modifying that instance into the next larger one. Commands used in the modification are recorded and then iterated to form larger family members. Thus, for example, the graph operations that transform the three node complete binary tree of Figure 2 into the seven node tree of Figure 3 can be repeated to form any larger, complete binary tree.[2] Figure 4 shows the full specification for the family of complete binary trees in which the appropriate iteration has been added by the programmer. This example also shows the use of labels; each node is labeled with its height (counting up from the leaves). We believe that many, if not all, useful graph families can be described in this iterative manner.

In our final system, we expect that input will be graphical and that the accompanying textual description will be produced automatically. For this reason, we have not been concerned with placement issues and we use a simple directional placement relative to *bounding boxes* (shown as rectangles in our figures). As nodes and subgraphs are added to the display, they are positioned using compass directions (`N`, `NE`, `E`, `SE`, `S`, `SW`, `W`, and `NW`) relative to the largest bounding box on the screen. This scheme is not entirely satisfactory. Consider, for example, the alternative description of a complete binary tree given in Figure 5. It builds the tree from the leaves upward by connecting a new root (added to the north) and appending a copy of the original tree to form the right (east) subtree. For this description, our placement strategy results in the tree becoming increasingly skewed as it grows. We expect that many of these problems will be solved when we switch to graphical input.

3. The Graph Assistant

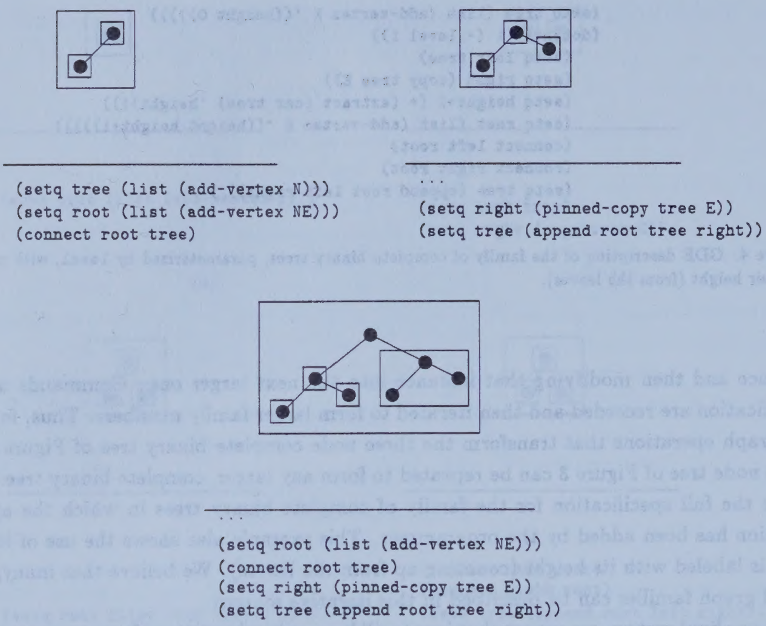
While our graph editor specifications contain the positional information needed for redisplay, it will often be necessary to display process structures for which this information is unavailable. Some process structures, for example, might be better described textually, others might vary dynamically.[3] In addition, tne user may want to view alternate displays of a single structure. To assist in these displays, the graph assistant supports the creation of aesthetic graph layouts.

The Graph Assistant uses heuristics for creating appealing displays that are invoked at the user's direction to place a graph on the screen. Figure 6 shows an example. The programmer entered his interconnection structure – a mesh – to the database and then invoked the Graph Assistant. A first attempt at a display is shown in Figure 6*a*. Interacting with the Graph

---

[2] Currently we use textual input but we expect to replace this with a graphical language in future versions of the editor.

[3] At this time, Simon does not allow dynamic creation or deletion of processes but we intend to add the necessary extensions.



```
(setq tree (list (add-vertex N)))
(setq root (list (add-vertex NE)))
(connect root tree)
```

```
...
(setq right (pinned-copy tree E))
(setq tree (append root tree right))
```

```
...
(setq root (list (add-vertex NE)))
(connect root tree)
(setq right (pinned-copy tree E))
(setq tree (append root tree right))
```

Figure 5: Tree built from the leaves upwards. In the first step, a root has been added to a single node using the `add-vertex` and `connect` operations; in the second step, a right subtree has been added using a `pinned` copy of the original (now left) subtree. In the third step, these steps are repeated to form the next family member.

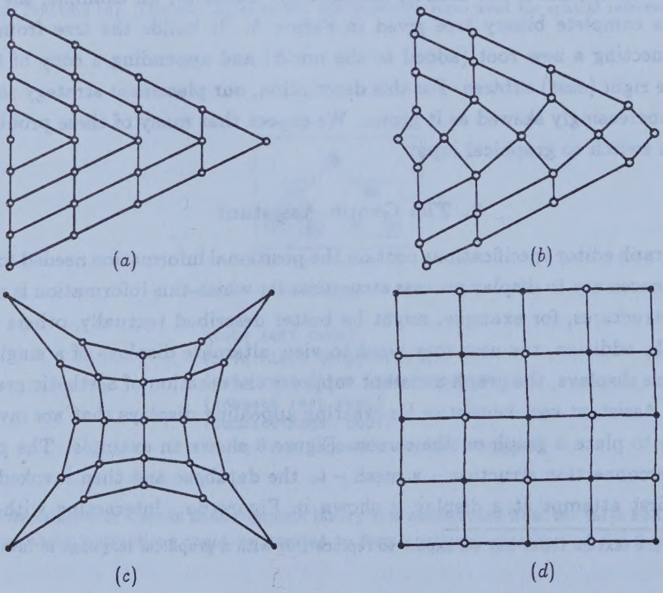

(a) (b) (c) (d)

Figure 6: Placement of the nodes of a mesh using the Graph Assistant.

Assistant, the user invoked a "pull" heuristic which minimizes the tension on the edges, to create the graph in Figure 6*b*; he then fixed the corners of the graph (shown by solid circles) and repeated the "pull" until the graph looked as shown in Figure 6*c*. The final display, Figure 6*d*, was created with the use of an "equalize" heuristic which creates edges with uniform lengths.

As another example, consider the cube connected cycle shown in Figure 7. It is shown as originally displayed in Figure 7*a* and after a single application of the "pull" heuristic in Figure 7*b*. At this point, the programmer manually "fixed" the nodes on the perimeter of the graph (as indicated by the filled in circles) and then iterated the "pull" heuristic to get the graph shown in Figure 7*c*. The final graph, Figure 7*d*, was achieved by manually separating the inner triangles (and repositioning the fixed nodes slightly to get the parallel cube edges).

We are currently investigating new layout heuristics, paying to particular attention to methods for exploiting symmetry.

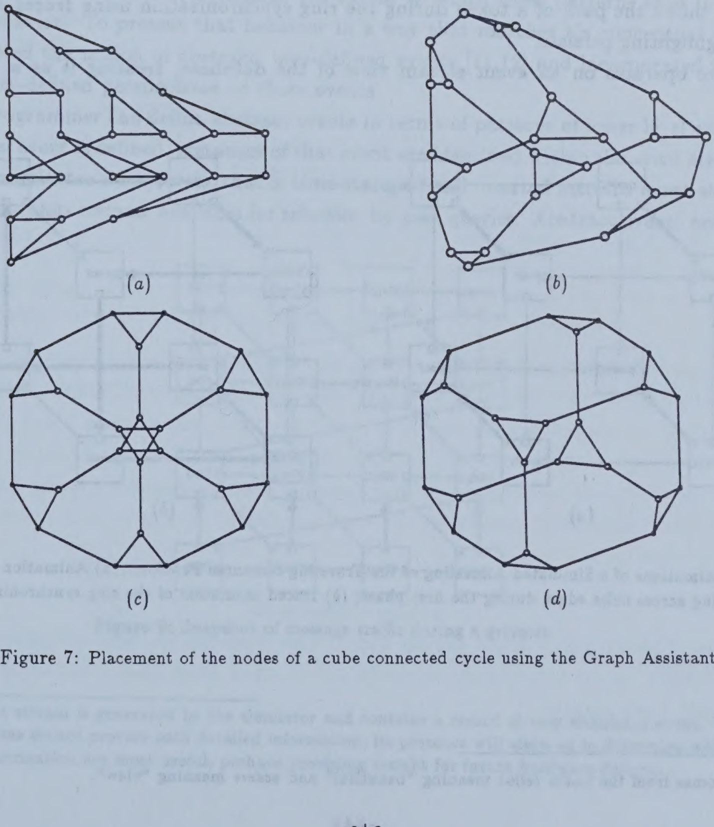


Figure 7: Placement of the nodes of a cube connected cycle using the Graph Assistant.

5. Parallel Debugging

Highly parallel computation is not amenable to existing debugging techniques since parallel programs do not have the consistent global states, manageable quantities of potentially relevant information or reproducibility that have formed the basis for sequential debugging paradigms. Instead, their behavior is best understood in terms of the flow of data and control resulting from interprocess communication. These behaviors are often very structured: fine grain, tightly coupled processes communicate across regular interconnection networks resulting, at least logically, in patterned data and control flows. Since many of the errors in parallel programs are communication related, we believe that the identification of these patterns will form the basis for highly parallel debugging paradigms.

We have designed and implemented a "pattern-oriented" debugger, called *Belvedere*[4] which we are using to investigate the animation and manipulation of interprocess communication patterns [9]. Belvedere is a trace-based, ***post-mortem*** debugger [6,7] providing animations of program behavior. Figure 8 shows snapshots of sample animations of the above two queries. The animations were taken from the trace of a hypercube program implementing simulated annealing of the traveling salesman problem [8]. Each iteration of the program has two phases; in the first, pairs of processes communicate along cube edges in order to evaluate possible perturbations and, in the second, a ring synchronization is performed to implement accepted changes. Animated events are shown with highlighting. Figure 8(*a*) shows the first phase of an iteration using normal animation in which highlighting lasts only as long as the event itself; Figure 8(*b*) shows the path of a token during the ring synchronization using ***traced*** animation in which highlighting persists.

Belvedere operates on an event stream view of the database, treating it as a relational

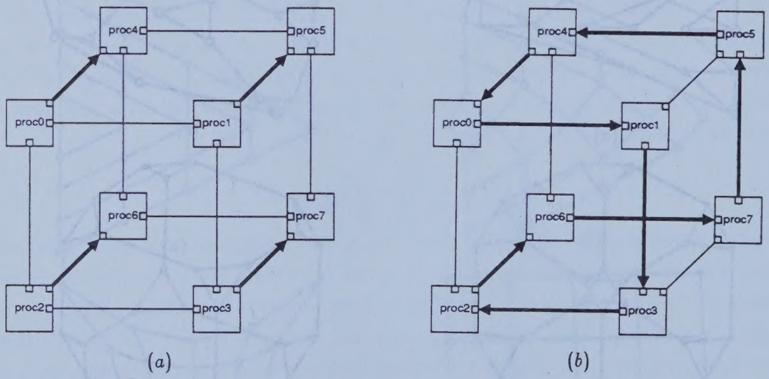


Figure 8: Animations of a Simulated Annealing of the Traveling Salesman Problem. (*a*) Animation of processes communicating across cube edges during the first phase; (*b*) traced animation of the ring synchronization.

---

[4] *Belvedere* comes from the Latin *bellus* meaning "beautiful" and *vedere* meaning "view".

database.[5] Users can select portions of the database for animation using standard database queries such as

```
    add messages with event=READ-MESSAGE and time<61400
```

and

```
    add messages with
             event=SEND-MESSAGE and message_name=conflicts
```

Each query results in the selection of a set of records which can be further modified by subsequent queries. At any time, the selected records can be animated. The above queries were used in the animations of Figure 8 (to get the animations as shown, with all of the processors and channels displayed, an additional set of background records were selected prior to each query).

While these simple animations are helpful, complex programs present patterns that are often difficult to interpret. Consider, for example, the snapshot of message traffic shown in Figure 9. It is taken from a *gridsort* program[10] in which an array of values (stored one per process) is sorted. During a single iteration, every process in the array does a comparison-exchange with each of its adjacent neighbors. Logically, these comparisons proceed in lock step: even rows, even columns, even diagonals, odd rows, odd columns, odd diagonals. In the snapshot, however, the pattern is obscured by low-level details and asynchronous execution. Since the programmer understands his code in terms of sequences of row, column, diagonal swaps, it is important that he be able to determine the extent to which these patterns exist in the actual system behavior. To present that behavior in a way that matches his conceptual models, we have adopted the notion of abstract, user-defined events [11,12] and incorporated facilities to impose user-defined perspectives on those events.

The programmer can define abstract events in terms of patterns of lower level events. Once a high-level event is defined, instances of that event are identified within the event stream. Each instance constitutes a new event that is time-stamped and inserted into the event stream. Abstract events thus become available for selection by user queries. Abstract events are animated

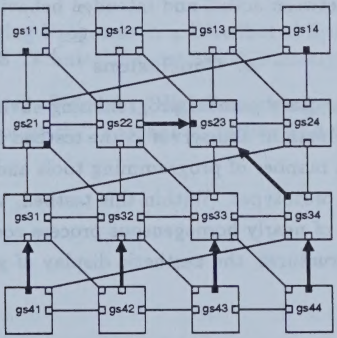
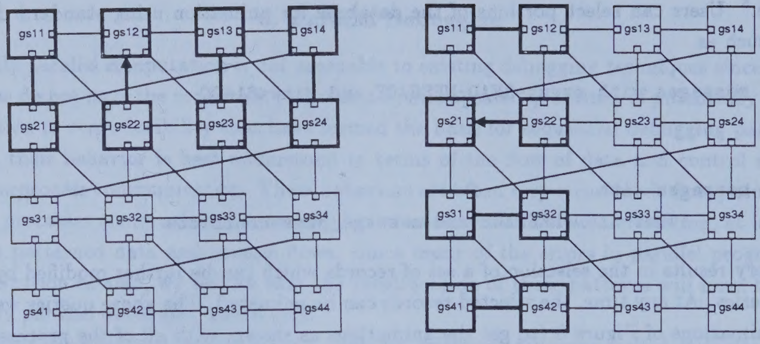


Figure 9: Snapshot of message traffic during a gridsort.

---

[5] The event stream is generated by the simulator and contains a record of very simulated event. While most existing systems do not provide such detailed information, its presence will allow us to determine which types of debugging information are most useful, perhaps providing insight for future hardware designs.



(*a*) (*b*)

Figure 10: High-level row (*a*) and column (*b*) events from the perspective of processor (2,2)

from user-defined points of reference or *perspectives*. The programmer could, for example, ask to see the gridsort's behavior from the perspective of processor (2,2) as shown in Figure 10. Note that he would then see the abstract row and column exchanges sequentially as he expected because processor (2,2) participates in those events sequentially. Belvedere supports perspectives from processors, channels and data items.

Viewing the behavior of a program from a single perspective may well be misleading: from the perspective of process $(1,1)$, for example, the user would never see any `ODD-ROW-SWAP`, `ODD-COLUMN-SWAP` or `ODD-DIAGONAL-SWAP` events. In addition, the use of perspectives often sequentializes concurrent behavior possibly obscuring the existence of timing errors. To gain accurate insights into his program's behavior, the user will have to view it from a variety of perspectives. Since this is easily done with Belvedere, we expect that it will be a valuable tool in determining discrepancies between actual and intended behaviors.

6. Conclusions

We have presented a rudimentary parallel programming environment that operates as the front end of a multiprocessor simulator and serves as the testbed for much of our research. The environment itself consists of a number of programming tools and a kernel which supports the development of new tools and prototypes. Within this testbed, we have begun to experiment with tools for the specification of nearly homogeneous process code, the explicit description of interprocess communication structures, the aesthetic display of graphs, and the debugging of parallel programs.

Acknowledgments

We would like to thank Mary Larson and Neville Newman for their contributions to the design of Simple Simon. In addition, we would like to thank Mark Bailey and Craig Loomis for their programming support.

References

[1] L. J. Osterweil, "Toolpack - An Experimental Software Development Research Project," *IEEE Transactions on Software Engineering* SE-9(6), pp.673-685 (1983).

[2] W.Teitelman and L.Masinter, "The Interlisp Programming Environment," *Computer*, pp.25-33 (1981).

[3] R. M. Fujimoto, "Simon: Simulator of Multicomputer Networks," Technical Report, Dept. of Electrical Engineering and Computer Science UCB/CSD 83/140, University of California at Berkeley (August 1983).

[4] D. E. Heller, "Multiprocessor Simulation Program Simon," Technical Report, Physics and Computer Science, Shell Development Corporation (November 1984).

[5] Duane A. Bailey and Janice E. Cuny, "An Approach to Programming Process Interconnection Structures: Aggregate Rewriting Graph Grammars," *Parallel Architectures and Languages Europe, Lecture Notes in Computer Science 259,* J.W. de Bakker, A.J. Nijman and P.C. Treleaven (eds.), Springer-Verlag, pp.112-123 (June 1987).

[6] R.M. Balzer, "EXDAMS - Extendible Debugging and Monitoring System," *Proceedings AFIPS Joint Spring Computer Conference,* pp. 567-580 (1969).

[7] H. Garcia-Molina, F. Germano, W.H. Kohler, "Debugging a Distributed Computing System," *IEEE Transactions on Software Engineering* SE-10(2), pp. 210-219 (March 1984).

[8] Edward Felten, Scott Karlin, and Steve W. Otto, "The Traveling Salesman Problem on a Hypercubic, MIMD Computer," Proceedings of the 1986 International Conference on Parallel Processing, pp. 6-10 (August 1986).

[9] Alfred A. Hough and Janice E. Cuny, "Belvedere: Prototype of a Pattern-Oriented Debugger for Highly Parallel Computation," *Proceedings of the 1987 International Conference on Parallel Processing*, pp. 735-738 (1987).

[10] Chip Weems, "Image Processings with a Content Addressable Array Parallel Processor," University of Massachusetts, COINS Technical Report 84-14 (September 1984).

[11] Peter C. Bates, "Debugging Programs in a Distributed System Environment," University of Massachusetts, COINS Technical Report 86-05 (January 1986).

[12] Peter C. Bates and Jack C. Wileden, "High-level debugging of distributed systems: the behavioral abstraction approach," *Journal of System Software* 3, pp. 255-244 (1983).

The Lattice Structure of a Bus Oriented Multiprocessor Model

A.A. Lazar and T.G. Robertazzi

Depts. of Electrical Engineering
Columbia University and SUNY at Stony Brook

Abstract

The geometry of the state transition lattice of a bus oriented multiprocessor model is described. It is shown that the lattice is comprised of an aggregation of lattices which are embedded into multidimensional toroidal manifolds. The connectivity hypergraph of these manifolds and the nature of their connectivity is described.

I. Introduction

To date most protocol modeling has been based on schematic network representations of protocol behavior such as queueing and Petri networks. While such schematic network representations are quite useful from the applied modeling point of view, it may be argued that the associated state transition lattice (diagram) more readily exposes the fine structure of models.

Markovian systems represent the simplest probabilistic model with an associated state transition lattice. Because of the memoryless property of the exponential distribution the state transition lattice contains all the information necessary to solve for the equilibrium probabilities of state. This information consists of the placement of transitions between states and the values of transition rate. For queueing networks with product form solution it was shown in 1984 [2] that the lattice (complex, in the language and topology) decomposes exactly into basic simplical building blocks (cells). Moreover the addition or removal of such building blocks does not change the product form solution, except for a possible renormalization. Modifications of non-product form protocols which lead to such an integral aggregation of building blocks also possess product form solution.

In 1986 a Duality Principle was described which made clear that the previously noted phenomena of local balance [7] in product form networks was actually the observation of the passage of an "isolated" cyclic flow of probability flux about the edges of a simplical building block and through a state. In essence, the circulatory structure of the product form network consists of an aggregation of isolated cyclic flows, each of which is geometrically embedded about the edges of a simplical region. The individual flows were also shown to correspond to certain sequences of queueing events.

There is also structure in certain non-product form networks. A class of non-product form networks whose equilibrium state probabilities can be determined in sub-groups [1,9] leads to large computational savings. Algebraic recursions are possible in many cases. This class is defined by a common geometric structure.

It was recently shown that there exists a class of Markovian Petri Networks which have a product form solution [3,4]. This was a surprising result as product form solutions had not been previously associated with stochastic Petri networks. The lattices are naturally embedded in multidimensional toroidal manifolds. This work demonstrates the importance of an understanding of the lattice geometry of state transition diagrams for probabilistic systems. In this paper we are concerned with variations on a multiprocessor model that originally appeared in [5]. It will be shown that the lattice is an amalgamation of component lattices embedded on multidimensional toroidal manifolds. The connectivity between these component manifolds is best described in terms of a connectivity hypergraph. For small multiprocessor systems the component manifolds may be embedded on certain common surfaces.

In section II the multiprocessor model is described. In section III the connectivity hypergraph and its properties are discussed.

II. A Markovian Multiprocessor Model

Consider a multiprocessor system, as shown in Fig. 1, consisting of two processors, P_1 and P_2 and the corresponding memories, M_1, and M_2. Our example is based on a slight modification of the TOMP architecture appearing in [5]. P_1 (or alternately, P_2) may attempt transfers to $M_1(M_2)$ through the local memory bus $MB_1(MB_2)$ connected to the private processor bus $PB_1(PB_2)$. Alternatively, $P_1(P_2)$ may attempt transfers to $M_2(M_1)$ via the global bus (GB) and the local memory bus $MB_2(MB_1)$. Conflicts occur as only one processor may utilize the local bus or the global bus at a time. Note that the processor memories, PM_1 and PM_2 , are only accessible from their respective processors on the private busses PB_1 and PB_2 , respectively. Unlike in [5], we assume that processor $P_1(P_2)$ attempts an access of the global bus only if the memory module $M_2(M_1)$ is free.

A Markovian Petri Net model of this system appears in Fig. 2. Formally, the MPN is defined as the sixtuple:

$$\mathbf{P} = (P , T , I , O , M , Q) \tag{2.1}$$

P is a set of places (drawn as circles) and T is a set of transitions (drawn as horizontal bars). I is an input function which maps each transition to one or more places (drawn as directed arcs) and O is an output function which maps each place to one or more transitions (drawn as continuous or dotted directed arcs). A marking, M, is an assignment of a non-negative number of tokens (drawn as dots) to individual places. For a given Petri Net, the marking defines the state of the system. Q is the set of rates associated with the transitions.

The rules of Petri Net operation are simple. If there is at least one token in each place incident to a transition, it is enabled for firing. This involves removing one token from each place incident to the transition (except from places connected with the transition with dotted arcs) and adding one token to each place connected to the transition by an outgoing arc. For a MPN the time between an enabled transition and its firing is an exponentially distributed random variable.

The class of MPN's considered in this paper is safe, that is, a place may hold at most one token at any given time. All transitions also exhibit liveness, *i.e.*, deadlock cannot occur.

In the multiprocessor MPN model of Fig. 2 one can see a total of four linear sequences of event originating from the places indicating that the processors are (internally) active. Each such sequence will be referred to as a *task sequence* and the states which comprise it as *sub-tasks.* The first state is often an idle state (*i.e.*, task waiting to be initiated). The four tasks of Fig. 2 thus correspond to P_1 or P_2 attempting to perform a transfer with M_1 or M_2

Generally, one can distinguish between places which represent sub-tasks and those which represent resources (*e.g.*, GB FREE). The essence of the Petri Net paradigm is that a task sequence may be blocked (temporarily or permanently) if resources are not available.

In Fig. 2, q^{kl} is the exponential rate associated with the l th sub-task (1 = 0,1,2,...) of the k th task sequence (k = 1,2,3,4). The Petri Net transitions should not be confused with the transitions of the state transition lattice of the Markov process associated with the Petri Net. The corresponding state transition lattice appears in Fig. 3. The horizontal axis corresponds to P_1 and task sequence 1 and 3 while the vertical axis corresponds to P_2 and task sequence 2 and 4. The equilibrium state probabilities, for each quadrant of the lattice, are given below for all states of non-zero probability. For all states (i, j) of the indicated task sequences we have:

$$p(i,j) = \frac{q^{10} q^{20}}{q^{1i} q^{2j}} p(0,0) \quad , \tag{2.2}$$

for tasks 1 and 2,

$$p(i,j) = \frac{q^{30} q^{40}}{q^{3i} q^{4j}} p(0,0) \quad ,$$

for tasks 3 and 4,

$$p(i,j) = \frac{q^{10} q^{40}}{q^{1i} q^{4j}} p(0,0) \quad ,$$

for tasks 1 and 4, and

$$p(i,j) = \frac{q^{20} q^{30}}{q^{2i} q^{3j}} p(0,0) \quad ,$$

for tasks 2 and 3.

Note that these probabilities have the characteristic product form which is known to exist for certain classes of queueing networks. The basis for this is discussed extensively in [3,4]. However because we will be discussing the connectivity of the quadrants of the state transition lattice for this, and generalized models, we do not need to assume a product form solution. That is, the structure of the state transition lattice may be modified within the quadrants so that a product form solution does not result.

III. The Connectivity of the Connectivity Hypergraph

Consider the state transition lattice of the multiprocessor shown in Fig. 3. The states and transitions of each quadrant can be embedded on a torus type space. These quadrants are pasted together along commonly shared states and transitions (Fig. 4). The overall connectivity can be explicitly demonstrated though a graph where each node represents a quadrant of the state transition lattice and edges indicate which quadrants are pasted together. For the embedding of Fig. 4 the resulting graph would have the form of a simple square.

What form would the connectivity graph take for more elaborate multiprocessor systems? Suppose there are m memories and n processors. Let the quadrant of interest be labeled as $M_a/M_b/M_c \;\; /M$ where the index in the i^{th} position indicates the memory to be accessed by processor P_i. The labels $M_a/M_b/M_c \;\; /M$ correspond to the individual nodes of the connectivity graph. The nodes are connected by hyperedges of degree m. Alternately each hyperedge connects m nodes. It should be noted that each quadrant consists of the many states and transitions associated with the specific set of memory accesses.

Figure 5 illustrates the connectivity graph for a multiprocessor with two processors and three memories. Note that the upper and lower edge can be naturally identified in this plane model of the space of embedding the connectivity hypergraph. The left and right sides can be identified in the opposing manner shown. Thus the connectivity hypergraph is naturally embedded in a Klein Bottle. On the other hand, if the four corner branches are rotated, the connectivity hypergraph can be embedded in a Moebius strip (Fig. 6).

The way in which quadrants are "pasted" to each other is of some interest. The pasting in its entirity in three dimensional space can be seen in Fig. 4 for the two processor, two memory case. This is not always possible. For the two processor, three memory case each quadrant can be toroidally embedded. If the tori are represented by plane models (see Fig. 3) the connectivity in a local region is illustrated in Fig. 7. The "edge" of each plane model is attached to the edges of two other plane models. The "edge" is actually a loop of the lattice (as in Fig. 3).

For a three processor, three memory model each quadrant is naturally embedded into a three dimensional torus type manifold. (A manifold is a finite space with no boundary.) The local connectivity is illustrated in Fig. 8. Notice how in passing through a cube face one passes into one of two cubic manifolds.

Fig. 9 illustrates the connectivity hypergraph for a two processor, four memory, model. If the upper and lower trees are rotated the lattice can be embedded in a two dimensional toroidal manifold (Fig. 10).

Some general points concerning the structure of the connectivity hypergraph can now be deduced:

Property 1: The connectivity hypergraph is symmetric with respect to all nodes and all hyperedges between nodes.

Property 2: For an m memory system the hyperedges are of degree m.

Property 3: The number of neighboring quadrants per quadrant node for n processors and m memories is n (m-1).

Property 4: Between two quadrant nodes $M_a/M_b/M_c \;\; /M$ and $M_r/M_s/M_t \;\; /M$ the length of the shortest path(s) between the two quadrant nodes is the Hamming distance, h, between their representation. For instance, the distance between 1/1/2 and 2/1/3 is two.

Property 5: Between two quadrant nodes there are h! paths of shortest length. This corresponds to h choices for the first processor to change the memory it is accessing, h-1 choices for the second and so on.

Property 6: The smallest cycle in the connectivity hypergraph is of length four. This corresponds to moving from $M_a/M_b/M_{c\ldots}/M$ to $M_r/M_s/M_{t\ldots}/M$, by changing the memories being accessed by two processors in one order and returning to $M_a/M_b/M_{c\ldots}/M$ in the same order.

IV. Conclusion

This paper has demonstrated that there is a very definite structure to the state transition lattices of multiprocessor Petri net models. The lattice is an aggregation of lattices embedded in torus type manifolds. The nature of the aggregation can be examined in terms of a connectivity hypergraph. Properties of the connectivity hypergraph can be deduced in a straight forward manner. However an open question involves the type of manifolds into which the lattice is embeddable for an arbitrary number of processors and memories.

Acknowledgment:

The research reported in this paper was supported in part by the SDIO/IST and managed by the U.S. Office of Naval Research under Grant No. N00014-85-K0610.

References

[1] Ferrandiz, J.M. and Lazar, A.A., "Geometric Analysis of Quasi-Birth-and- Death Processes by Flow Redirection", *Proc. of the 1987 Conf. on Info. Sci. and Sys.*, John Hopkins University, Baltimore, Md., March 1987.

[2] Lazar, A.A. and Robertazzi, T.G., "The Geometry of Lattices for Multiclass Markovian Queueing Networks", *Proc. of the 1984 Conf. on Info. Sci. and Sys.*, Princeton University, Princeton N.J., March 1984, pp. 164-168.

[3] Lazar, A.A. and Robertazzi, T.G., "The Algebraic and Geometric Structure of Markovian Petri Network Lattices", *Proc. of the Twenty-Fourth Ann. Allerton Conf. on Comm., Control and Comp.*, Univ. of Illinois, Urbana-Champaign, Ill., Oct. 1986.

[4] Lazar, A.A. and Robertazzi, T.G., "Markovian Petri Net Protocols with Product Form Solution", *IEEE INFOCOM*, San Francisco CA., March 1987.

[5] Marsan, M.J., Balbo, G. Conte, G. and Gregoretti, F., "Modeling Bus Contention and Memory Interference in a Multiprocessor System", *IEEE Trans. on Computers*, Vol. C-32, No. 1, Jan. 1983, pp. 60-72.

[6] Peterson, J.L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, 1981.

[7] Chandy, K.M., Howard, J.H. and Towsley, D.F., "Product Form and Local Balance in Queueing Networks", *JACM*, Vol. 24, No. 3, April 1977, pp. 250-263.

[8] Wang, I.Y. and Robertazzi, T.G., "The Probability Flux Circulation of Certain Communication and Computation State Models", *Proc. of the 1986 Conf. on Info. Sci. and Sys., Princeton University, Princeton N.J., March 1986.*

[9] Wang, I.Y. and Robertazzi, T.G., "Recursive Computation of Steady State Probabilities of Non-Product Form Queueing Models Associated with Computer Network Protocols", *30th Midwest Symposium on Circuits and Systems*, Syracuse N.Y., Aug. 1987.

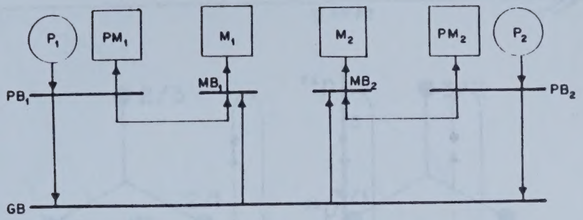

P1
PM1
M1
M2
PM2
P2
PB1
MB1
MB2
PB2
GB

Fig. 1

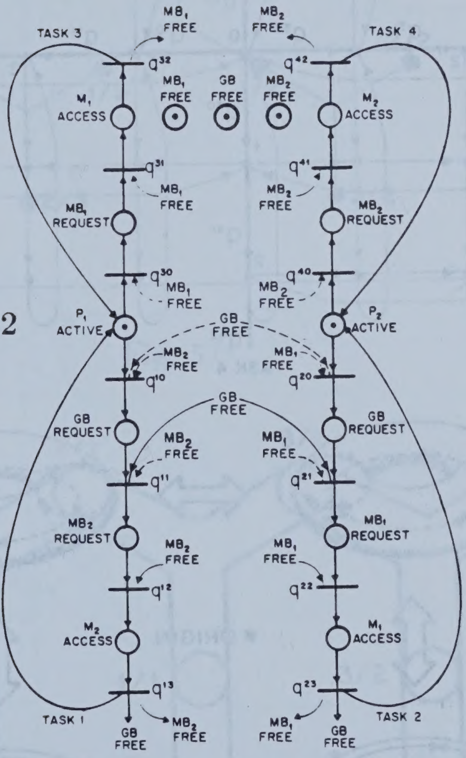

TASK 3
MB1 FREE
MB2 FREE
TASK 4
q32
q42
M1 ACCESS
MB1 FREE
GB FREE
MB2 FREE
M2 ACCESS
q31
q41
MB1 FREE
MB2 FREE
MB1 REQUEST
MB2 REQUEST
q30
q40
MB1 FREE
MB2 FREE
P1 ACTIVE
GB FREE
P2 ACTIVE
MB2 FREE
MB1 FREE
q10
q20
GB FREE
GB REQUEST
GB REQUEST
MB2 FREE
MB1 FREE
q11
q21
MB2 REQUEST
MB1 REQUEST
MB2 FREE
MB1 FREE
q12
q22
M2 ACCESS
M1 ACCESS
q13
q23
TASK 1
GB FREE
MB2 FREE
MB1 FREE
GB FREE
TASK 2

Fig. 2

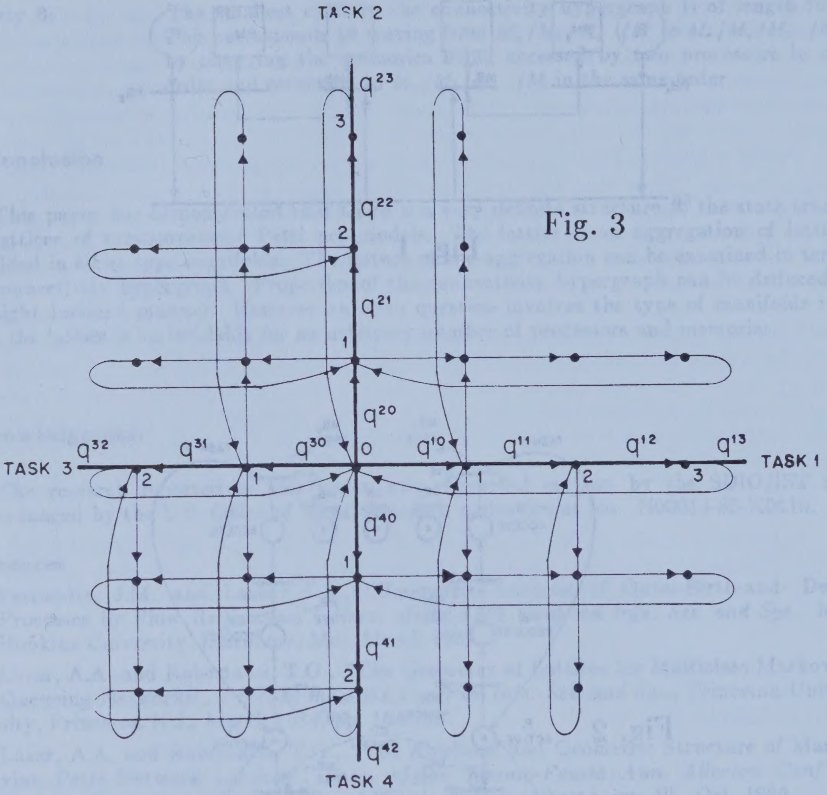

TASK 2
q23
3
q22
2
q21
1
q20
q32
q31
q30
0
q10
q11
q12
q13
TASK 3
TASK 1
2
1
1
2
3
q40
1
q41
2
q42
TASK 4

Fig. 3

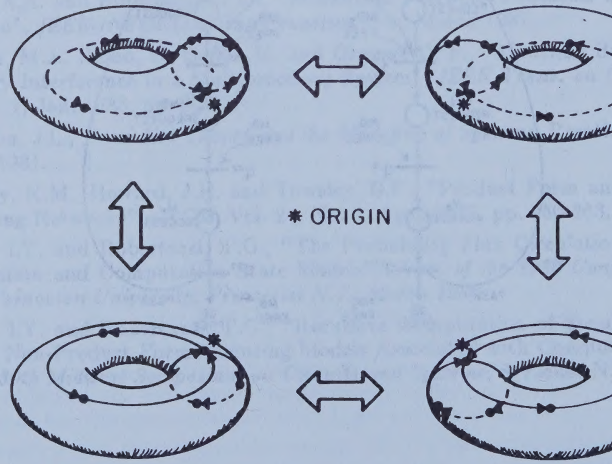

* ORIGIN

Fig. 4

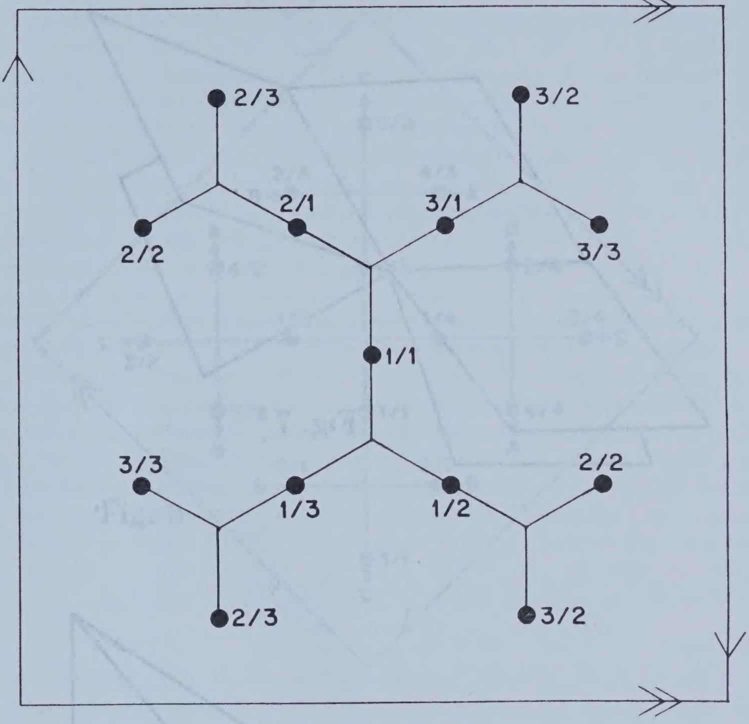

2/3
3/2
2/1
3/1
2/2
3/3
1/1
3/3
2/2
1/3
1/2
2/3
3/2

Fig. 5

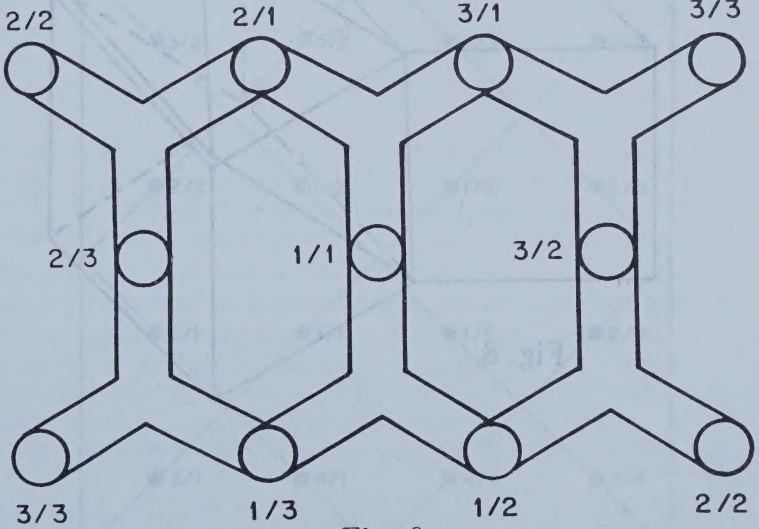

2/2
2/1
3/1
3/3
2/3
1/1
3/2
3/3
1/3
1/2
2/2

Fig. 6

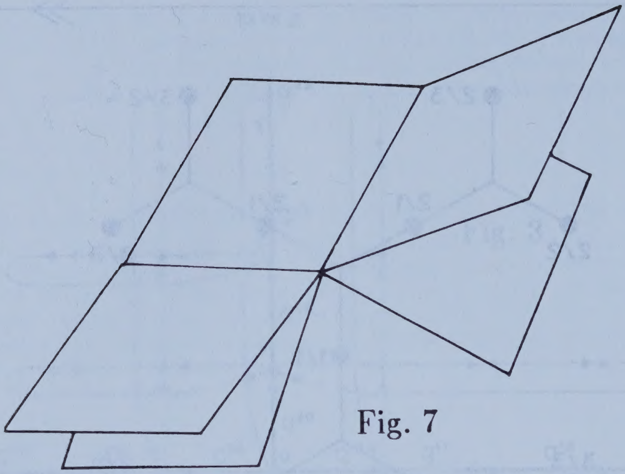


Fig. 7

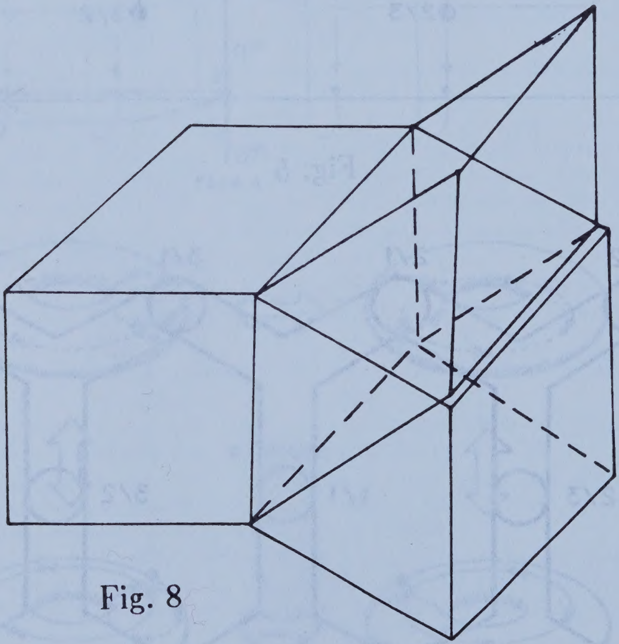


Fig. 8

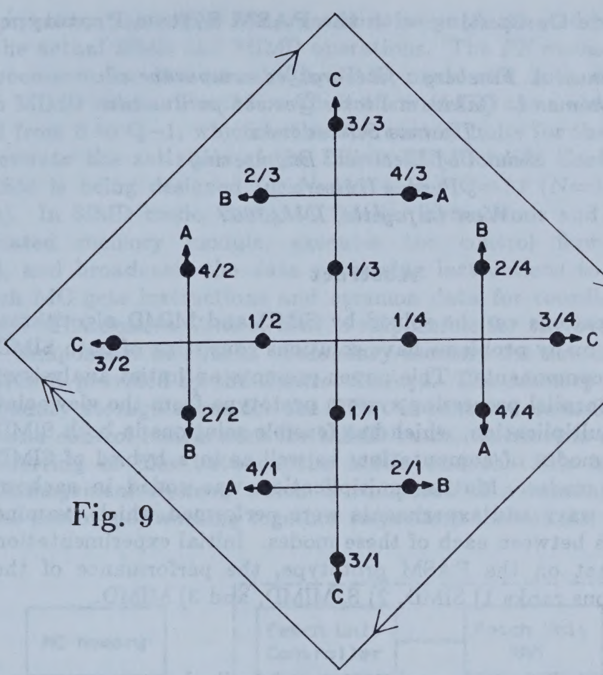
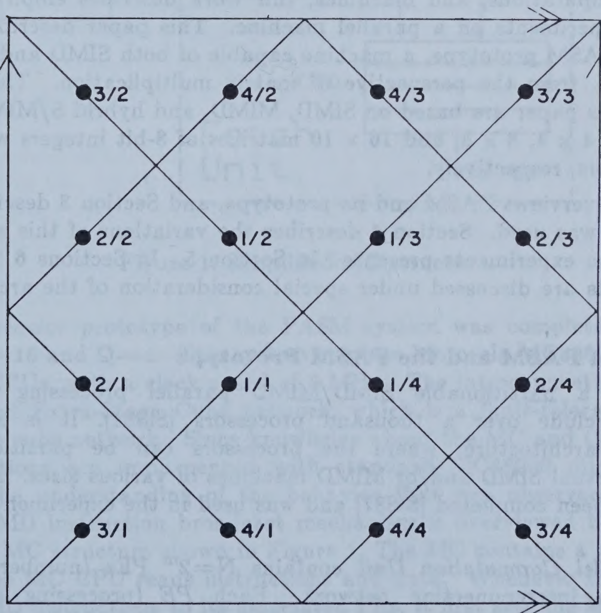

C
3/3
2/3
B
4/3
A
A
4/2
1/3
B
2/4
1/2
1/4
3/4
C
C
3/2
2/2
1/1
4/4
B
A
4/1
A
2/1
B
3/1
C

Fig. 9

Fig. 10


3/2
4/2
4/3
3/3
2/2
1/2
1/3
2/3
2/1
1/1
1/4
2/4
3/1
4/1
4/4
3/4

Mixed-Mode Computing with the PASM System Prototype

Samuel A. Fineberg : fineberg@ed.ecn.purdue.edu
Thomas L. Casavant : tomc@ee.ecn.purdue.edu
Thomas Schwederski
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

Abstract

Many applications can be solved by SIMD and MIMD algorithms. In addition, many problems have solutions consisting of both SIMD and MIMD components. This paper presents an initial analysis of the PASM parallel processing system prototype from the viewpoint of matrix multiplication, which has feasible solutions in both SIMD and MIMD modes of computation, as well as in a hybrid of SIMD and MIMD modes. Matrix multiplication was coded in each of these three ways and experiments were performed which examine the tradeoffs between each of these modes. Initial experimentation indicates that on the PASM prototype, the performance of the three solutions ranks 1) SIMD, 2) S/MIMD, and 3) MIMD.

1. Introduction

Two of the parallel machine architectures described in a taxonomy by Flynn [Fly66] are SIMD (single instruction stream – multiple data streams) and MIMD (multiple instruction streams – multiple data streams). While much theoretical work in the past has been done with respect to performance of SIMD and MIMD algorithms, computations, and machines, this work describes empirical data generated from experiments on a parallel machine. This paper describes the performance of the PASM prototype, a machine capable of both SIMD and MIMD modes of computation, from the perspective of matrix multiplication. The experiments described in this paper are based on SIMD, MIMD, and hybrid S/MIMD algorithms for multiplying 4×4, 8×8, and 16×16 matrices of 8-bit integers which use 4, 8, and 16 processors, respectively.

Section 2 overviews PASM and its prototype, and Section 3 describes the basic algorithm that was used. Section 4 describes the variations of this algorithm used to carry out the experiments presented in Section 5. In Sections 6 through 9, the empirical results are discussed under special consideration of the architecture that was employed.

2. Overview of PASM and the PASM Prototype

PASM is a partitionable SIMD/MIMD parallel processing system being designed to include over a thousand processors [SiS81]. It is a dynamically reconfigurable architecture, where the processors can be partitioned to form independent virtual SIMD and/or MIMD machines of various sizes. A 30-processor prototype has been completed [SiS87] and was used in the experiments described in this paper.

The *Parallel Computation Unit* contains $N=2^m$ PEs (numbered from 0 to N–1), and an interconnection network. Each *PE* (processing element) is a processor/memory pair. The *PE processors* are sophisticated microprocessors that perform the actual SIMD and MIMD operations. The *PE memory modules* are used by the processors for data storage in SIMD mode and both data and instruction storage in MIMD mode. The *Micro Controllers (MCs)* are a set of $Q=2^q$ processors, numbered from 0 to $Q-1$, which act as the control units for the PEs in SIMD mode and orchestrate the activities of the PEs in MIMD mode Each MC controls N/Q PEs. PASM is being designed for N=1024 and Q=32 (N=16 and Q=4 in the prototype). In SIMD mode, each MC fetches instructions and common data from its associated memory module, executes the control flow instructions (e.g., branches), and broadcasts the data processing instructions to its PEs. In MIMD mode, each MC gets instructions and common data for coordinating its PEs from its memory. The *System Control Unit* is responsible for the overall coordination of the other components of PASM. Secondary memory for the System Control Unit and the MCs is provided by the *Control Storage*. The *Memory Storage System* provides secondary storage space for the PEs. It contains data files for SIMD mode, and data and control instructions for MIMD mode. It consists of N/Q independent The transferring of files between the MSUs and the PEs is controlled by the *Memory Management System,* which is composed of a separate set of processors (four in the prototype) working together to perform their tasks.

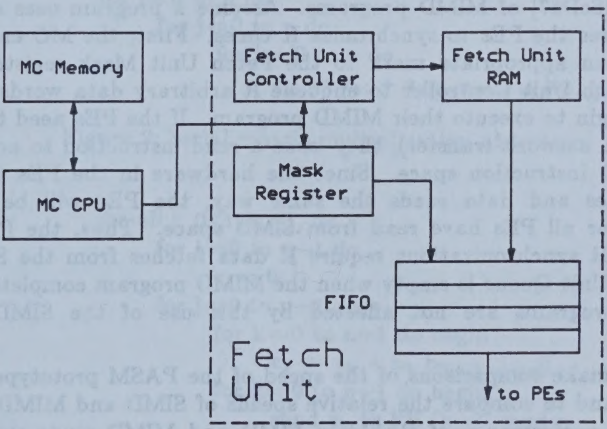


Figure 1: Simplified MC structure.

A 30 processor prototype of the PASM system was completed in December 1986, with N=16 and Q=4. This system employs Motorola MC68000 processors as PE and MC CPUs, with a clock speed of 8 MHz. The interconnection network is a circuit-switched Extra-Stage Cube network, which is a fault-tolerant variation of the multistage cube network. Since knowledge about the MC and the way in which SIMD instructions are implemented with standard MC68000 microprocessors is essential to the understanding of the behavior that was observed in the experiments, the SIMD instruction broadcast mechanism is overviewed below. Consider the simplified MC structure shown in Figure 1. The MC contains a memory module from which the MC CPU reads instructions and data. Whenever the MC needs to broadcast SIMD instructions to its associated PEs, it first sets the Mask Register in the Fetch Unit, thereby determining which PEs will participate in the following instructions. It then writes a control word to the Fetch Unit Controller which specifies the location and size of a block of SIMD instructions in the Fetch Unit RAM. The Fetch Unit Controller automatically moves this block word by word into the Fetch Unit Queue. Whenever a instruction word is enqueued, the current value of the mask register is enqueued as well. Since the Fetch Unit enqueues blocks of SIMD instructions automatically, the MC CPU can proceed with other operations without waiting for all instructions to be enqueued.

PEs execute SIMD instructions by performing an instruction fetch from a reserved memory area called the SIMD instruction space. Whenever logic in the PEs detects an access to this area, a request for an SIMD instruction is sent to the Fetch Unit. Only after all PEs that are enabled for the current instruction have issued a request is the instruction released by the Fetch Unit FIFO, and the enabled PEs receive and execute the instruction. Disabled PEs do not participate in the instruction and wait until an instruction is broadcast for which they are enabled. This way, switching from SIMD and MIMD mode is reduced to executing a jump instruction to the reserved memory space, and a switch from MIMD to SIMD mode is performed by sending a jump to the appropriate PE MIMD instruction address located in the PE main memory space.

The SIMD instruction broadcast mechanism can also be utilized for barrier synchronization [ScD87] of MIMD programs. Assume a program uses a single MC group, and requires the PEs to synchronize R times. First, the MC enables all its PEs by writing an appropriate mask to the Fetch Unit Mask register. Then it instructs the Fetch Unit Controller to enqueue R arbitrary data words, and starts its PEs which begin to execute their MIMD program. If the PEs need to synchronize (e.g., before a network transfer), they issue a read instruction to access a location in the SIMD instruction space. Since the hardware in the PEs treats SIMD instruction fetches and data reads the same way, the PEs will be allowed to proceed only after all PEs have read from SIMD space. Thus, the PEs are synchronized. The R synchronizations require R data fetches from the SIMD space. Thus, the Fetch Unit Queue is empty when the MIMD program completes, and subsequent SIMD programs are not affected by this use of the SIMD broadcast mechanism.

In order to make comparisons of the speed of the PASM prototype relative to other machines and to compare the relative speeds of SIMD and MIMD instruction fetches, the raw performance of PASM in SIMD and MIMD mode was measured and is illustrated in Table 1 in MIPS (millions of integer instructions per second) for two different types of instructions. The difference in speed between SIMD and MIMD modes can be attributed to two factors. SIMD instructions are fetched from the Fetch Unit Queue in the MC, and the queue can deliver data with one less wait state than can the PEs' main memories. In addition, PEs' main memories are implemented with dynamic memories. While care was taken in the hardware design that all refresh operations occur simultaneously in all PEs, and are performed invisible to the PE CPU, some delay is still possible. No such delay occurs during SIMD instruction fetches since the Fetch Unit queue is implemented with static RAM components.

Table 1: Prototype Raw Performance

Mode	Operation	Processing Rate
SIMD	16-bit Reg.-to-Reg. add	22 MIPS
MIMD	16-bit Reg.-to-Reg. add	18 MIPS
SIMD	16-bit Reg.-to-Mem. add	6.4 MIPS
MIMD	16-bit Reg.-to-Mem. add	6.0 MIPS

3. Matrix Multiplication Algorithms Used

The matrix multiplication algorithms used in this research were $O(n^2)$ time complexity algorithms which multiply two square matrices, A and B, and produce a resultant C matrix employing n PEs. Figure 2 shows an $O(n^3)$ serial algorithm, while Figure 3 illustrates a functionally equivalent parallel version. Note that the index *i* of the outer *for all* loop represents iteration across processors in space rather than iteration in time. In the parallel algorithm, the indices were permuted such that all three matrices A, B, and C can be stored in column format.

```
for i=0 to n do
    for j=0 to n do
        for k=0 to n do
            $c_{ki} = c_{ki} + a_{k((i+j) \bmod n)} b_{((i+j) \bmod n)i}$;
```

Figure 2: Serial matrix multiplication algorithm.

```
for all i, $0 \leq i \leq n-1$, do
    for j=0 to n−1 do
        $c_j = 0$;
    for j=0 to n−1 do
        for k=0 to n−1 do begin
            $c_k = c_k + a_k b_{((i+j) \bmod n)}$;
        for k=0 to n−1 do begin
            send $a_k$ to PE (k−1) mod n
            receive a value and move it into $a_k$
```

Figure 3: Parallel matrix multiplication algorithm.

This column format was chosen so that the data layout was uniform for all matrices. Three reasons for the uniformity exist. Since all matrices have the same storage format, a multiplication of B×A is possible without any data movement. Second, each matrix can be used in subsequent matrix multiplications without reformatting, and finally, the uniformity was desirable to facilitate parallel I/O from large data sets stored on secondary memory. The columnar storage gives the algorithm an O(n) space complexity per PE. The data layout (for n=4) is pictured in Figure 4.

PE 0			•••	PE 3		
a_{00}	b_{00}	c_{00}		a_{03}	b_{03}	c_{03}
a_{10}	b_{10}	c_{10}	•••	a_{13}	b_{13}	c_{13}
a_{20}	b_{20}	c_{20}		a_{23}	b_{23}	c_{23}
a_{30}	b_{30}	c_{30}		a_{33}	b_{33}	c_{33}

Figure 4: Data allocation for n=4

Now we consider how the multiplication progresses in detail. During each of the n iterations of the first innermost loop shown in Figure 3, each of the elements of the A matrix are multiplied by an element of the B matrix and added to the proper element of the C matrix for a total of n multiplications and n additions. In the second innermost loop, the columns of the A matrix are rotated to PE (k minus 1) mod n. These data transfers require n network transfer operations per PE (one for each element in the column). This procedure is repeated until all of the columns of the A matrix have been in all of the PEs for a total of n repetitions. The first two iterations (for n=4) carried out in PE 0 are pictured in Figure 5.

Due to the permutation capability of the PASM network, the operation required to perform the exchange allows all n PEs to send to PE (k minus 1) mod n in parallel. However, no block transfers are possible, and so each element must be transmitted separately.

The PASM prototype network is circuit switched, and setting up a path or dropping it is a time-consuming operation. In the matrix multiplication, only a single path setting is required, and thus the path was kept open once it was established. Hence, there were n^2 network accesses, n^2 multiplications, and $2n^2$ additions required, resulting in an $O(n^2)$ growth in execution time as n increases.

4. Variations on the Algorithm

Three variations of this basic algorithm were programmed in MC68000 assembly language and executed on the PASM prototype. A pure SIMD, pure MIMD, and an S/MIMD algorithm that employs SIMD-type barrier synchronization were coded for 4, 8 and 16 processors.

4.1. SIMD

There were three major factors that made the SIMD version of the algorithm the most natural choice. First, both arithmetic and network operations are data independent with the exception of the multiplication operations which are the same on all PEs but whose execution time is variable. This eliminated the need for setting masks to enable and disable PEs and eliminated the need for PE to MC communication. Second, the implicit synchronization inherent in SIMD mode allowed us to carry out the network transfer operations in a straightforward fashion requiring only three machine instructions. One final advantage of the SIMD version was that the use of the FIFO queue to buffer the instructions being sent to the PEs allowed the instructions being sent to the PEs to be overlapped with the execution of control flow instructions in the MCs.

Besides these conceptual factors involved in the SIMD version, there are some factors that were present due to the actual implementation of the PASM system prototype. First, instructions may be accessed at a faster rate from the fetch unit

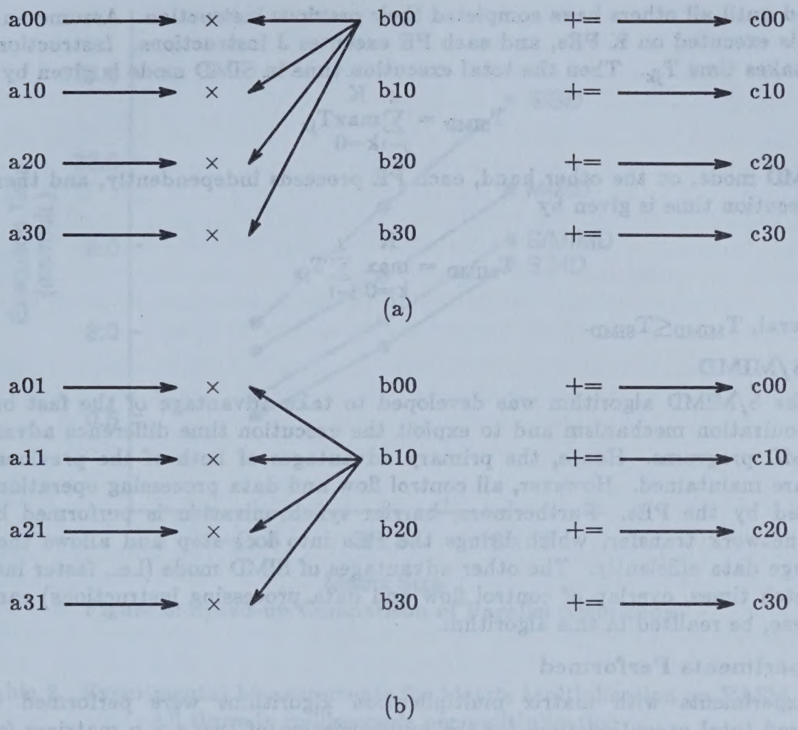


Figure 5: First two iterations in PE 0. (a) $j=0$, $0 \leq k \leq 3$. (b) $j=1$, $0 \leq k \leq 3$.

queue than from the PE memory. This is due to the actual memories used to implement these devices. Also, the overlap of the control flow instructions with the PE instructions is dependent on the fetch unit queue remaining non-empty. If the MC is not able to fill the FIFO at a rate faster than the PEs can take data from it then the PEs will not be able to proceed until the MC is able to put more instructions in the queue.

4.2. MIMD

The second version was a pure MIMD algorithm, in which the MCs were used only for initiating the matrix multiplications. The PEs executed the network accesses and all arithmetic operations asynchronously. The asynchronous data transmission necessitated polling of the network buffer in order to determine whether it was ready to accept data during send operations and whether data was valid after a receive operation. This process was very time-consuming due to the software overhead required to check the buffers and wait if necessary.

One advantage of the MIMD algorithm, however, was rooted in variations of instruction execution times. In the MC68000, not all instructions require the same amount of time, and in addition, some instructions (e.g., integer multiply and divide operations) have data dependent execution times. In SIMD mode, no PE can proceed until all others have completed their previous instruction. Assume an algorithm is executed on K PEs, and each PE executes J instructions. Instruction j on PE k takes time T_{jk}. Then the total execution time in SIMD mode is given by

$$T_{SIMD} = \sum_{j=1}^{J} \max_{k=0}^{K} T_{jk}$$

In MIMD mode, on the other hand, each PE proceeds independently, and therefore the execution time is given by

$$T_{MIMD} = \max_{k=0}^{K} \sum_{j=1}^{J} T_{jk}$$

In general, $T_{MIMD} \leq T_{SIMD}$.

4.3. S/MIMD

The S/MIMD algorithm was developed to take advantage of the fast barrier synchronization mechanism and to exploit the execution time difference advantage of MIMD programs. Hence, the primary advantages of both of the previous versions are maintained. However, all control flow and data processing operations are executed by the PEs. Furthermore, barrier synchronization is performed before every network transfer, which brings the PEs into lock-step and allows them to exchange data efficiently. The other advantages of SIMD mode (i.e., faster instruction fetch times, overlap of control flow and data processing instructions) can not, of course, be realized in this algorithm.

5. Experiments Performed

Experiments with matrix multiplication algorithms were performed which measured total execution time for the multiplication of two n × n matrices for the SIMD, MIMD, and S/MIMD modes with n = 4, 8, and 16. All operations were 8-bit integer operations, and overflow was ignored. The experimental results obtained were derived from the execution times for 4,096 iterations of each algorithm on the initial data. Measurements were made using the system's internal timers (Motorola 68230) and represent the average of three trials divided by 4096. Experiments were performed for each version of the algorithm on three sets of data:

(i) The Identity Matrix (in A) times the Identity Matrix (in B),

(ii) The Identity Matrix (in A) times Random data (in B), and

(iii) The Identity Matrix (in A) times Constant data ($FF) (in B).

The data value of the destination operand in the MC68000 multiply instruction does not affect the execution time. The elements of the A matrix were always used as destination operands, and thus the identity matrix could be chosen for A without affecting results. By using the identity matrix, matrix multiplication results could easily be verified, thereby simplifying the debugging process.

Experiment (i) represents an approximation of best case analysis for the multiplication times, (ii) represents the average case, and (iii) is worst case. The time required for each of these experiments appears in milliseconds in Table 2.

6. Speed-up & Overall Comparison

Figure 6 illustrates the speed-up obtained in the parallel versions of the algorithm. Clearly indicated is the $O(n^3)$ to $O(n^2)$ reduction in complexity from SISD to each of the parallel versions as well as the relative speed-ups of the three parallel versions.

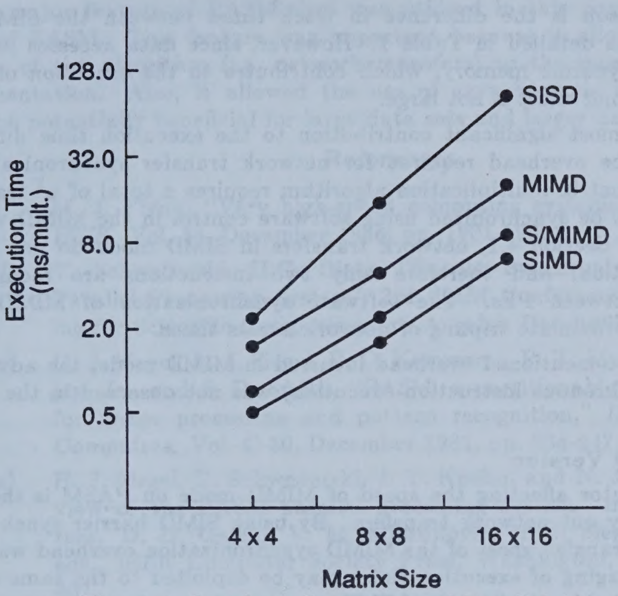


Figure 6: Speed-up Comparison of Parallel Algorithms.

Table 2. Experimental Measurements for Matrix Multiplication on PASM. All times in milliseconds per multiplication

Size	Mode	× I	× Random	× $FF
4×4	SIMD	0.50	0.52	0.53
	S/MIMD	0.64	0.66	0.67
	MIMD	1.42	1.45	1.45
8×8	SIMD	1.76	1.84	1.88
	S/MIMD	2.44	2.53	2.57
	MIMD	5.27	5.33	5.39
16×16	SIMD	6.42	6.86	6.98
	S/MIMD	8.01	8.40	8.52
	MIMD	20.24	20.51	20.75

7. SIMD vs. MIMD

The most predominant feature of the data in Table 2 is the difference between the SIMD and MIMD versions. There are three primary reasons for these differences. The first is the overlap of flow-control (in the MCs) and data processing (in the PEs) instructions in SIMD mode. The flow-control instructions consist of several branch instructions and JSR operations needed to orchestrate the actions of the PEs and to repeat the multiply and rotate operations n times. In MIMD mode, these operations are not performed in parallel with the data processing operations.

The second reason is the difference in fetch times between the SIMD and MIMD instructions as detailed in Table 1. However, since data accesses in SIMD mode are done to dynamic memory, which contributes to the *execution* of SIMD instructions, this second effect is not large.

The third and most significant contribution to the execution time difference between modes is the overhead required for network transfer synchronization in MIMD mode. The matrix multiplication algorithm requires a total of n^2 transfers. These transfers must be synchronized using software control in the MIMD version. As was discussed in Section 4.1, network transfers in SIMD mode do not require explicit synchronization, and therefore only two instructions are required to exchange a byte between PEs. The software synchronization of MIMD mode accounted for an approximate tripling of network access times.

Due to the above-mentioned overhead incurred in MIMD mode, the advantage of MIMD (i.e., asynchronous instruction execution) was not observed in the results presented here.

8. Hybrid S/MIMD Version

The greatest factor affecting the speed of MIMD mode on PASM is the overhead needed to carry out network transfers. By using SIMD barrier synchronization following each transfer, most of the MIMD synchronization overhead was eliminated, and the averaging of execution times may be exploited to the same extent. This hybrid version is identical to the MIMD program except in the way network transfers are handled, and thus has the the same data dependency characteristics as the MIMD program. If data dependency effects are large enough, they may outweigh the loss of overlap of control and data processing instructions of the SIMD version. While, in the measured data, it is seen that the hybrid version was much closer in performance to the SIMD than to the MIMD version, the data dependency was not sufficient to outperform the SIMD algorithm. This was due in part to the use of 8-bit integers for the matrix multiplications. While tests with 8-bit integers showed only a 10% variation in multiplication time for best and worst case, this variation may be as much as 78% for 16-bit quantities. Future experimentation will be directed at 16-bit algorithms, where it is expected that S/MIMD outperforms SIMD matrix multiplication algorithms.

It is also worthy of note that the MIMD version was closer in speed to SIMD mode than to MIMD mode. This indicates that the time required by the synchronization instructions in MIMD mode was the biggest contribution to the time required by of the MIMD program. In fact, the only difference between the MIMD and the hybrid version was the method of carrying out the transfer. This factor shows that, on PASM, the use of this type of synchronization allows performing tasks with speed very close to that of SIMD mode while decoupling asynchronous computations.

9. Conclusions

We achieved an O(n) speedup for matrix multiplication by implementing parallel versions of a matrix multiplication algorithm. Also, we were able to show that $T_{SIMD} < T_{S/MIMD} < T_{MIMD}$ for our matrix multiplication algorithm versions. One of the other features of PASM that was demonstrated was the ability of PASM to overlap control flow instructions in the MCs with data operations in the PEs when operating in SIMD mode.

A major feature of PASM that was utilized in this paper was the S/MIMD mode of PASM. This feature was important because it allowed tailoring certain aspects of the algorithm (i.e. network transfers) to the most natural method of implementation. Also, it allowed the use of asynchronous multiplications which could be potentially beneficial for large data sets and larger data values.

References

[Fly66] M. J. Flynn, "Very high-speed computing systems," *Proceedings of the IEEE,* Vol. 54, December 1966, pp. 1901-1909.

[ScD87] T. Schwederski, H.G. Dietz, "Barrier Synchronization in the PASM Parallel Processing System", 3rd SIAM Conference on Parallel Processing for Scientific Computing, Los Angeles, Dec. 1987.

[SiS81c] H. J. Siegel, L. J. Siegel, F. C. Kemmerer, P. T. Mueller, Jr., H. E. Smalley, Jr., and S. D. Smith, "PASM: a partitionable SIMD/MIMD system for image processing and pattern recognition," *IEEE Transactions on Computers,* Vol. C-30, December 1981, pp. 934-947.

[SiS87a] H. J. Siegel, T. Schwederski, J. T. Kuehn, and N. J. Davis IV, "An overview of the PASM parallel processing system," in *Computer Architecture,* D. D. Gajski, V. M. Milutinovic, H. J. Siegel, and B. P. Furht, eds., IEEE Computer Society Press, Washington, D.C., 1987, pp. 387-407.

[Sto80] H. S. Stone, "Parallel computers," in *Introduction to Computer Architecture (second edition),* H. S. Stone, ed., Science Research Associates, Inc., Chicago, IL, 1980, pp. 363-425.

[SiD87] T. Schwederski, H. G. Dietz, "Barrier Synchronization in the PASM Parallel Processing System," 3rd SIAM Conference on Parallel Processing for Scientific Computing, Los Angeles, CA, Dec. 1987.

ONE-TO-K COMMUNICATION IN DISTRIBUTED-MEMORY MULTIPROCESSORS[†]

Hyeong-Ah Choi, Abdol-Hossein Esfahanian, and ***Lionel M. Ni***
Department of Computer Science
Michigan State University
East Lansing, Michigan 48824-1027

1. INTRODUCTION

Distributed-memory multiprocessor (DMM) have demonstrated to be the least expensive method to establish a parallel processing environment [HMSC86, GrRe86, Seit85]. In such a system, which may be composed of hundreds or even thousands of processors, each processor has its own memory, and message passing is the only means for information exchange between processors. Both dynamic and static interconnection networks may serve as a communication subsystem for inter-processor communication. This paper considers DMMs with static interconnection topology, and hereafter DMM will exclusively refer to such systems.

In a DMM, each processor is directly connected to a small number of other processors (called *neigboring* processors). Neighboring processors can communicate directly, whereas non-neighboring processors have to communicate indirectly through other processors. Thus, in a DMM, a massage, in general, may have to go through several intermediate processors before reaching its destination. As indicated in [Foxg83], interprocessor communication is an important factor affecting the efficiency of a DMM. The message handling strategies and techniques, and how they are implemented, constitute the effectiveness of interprocessor communication. This paper studies interprocessor communication issues that arise in DMMs. Graphs will be used to model the underlying topology of DMMs; nodes will represent processors and edges will represent the bidirectional communication links between processors.

A *one-to-k* communication in a DMM refers to the situation in which a processor (called the *source*) wants to send a message to k, $k \geq 1$, other processors (*destinations*). Clearly, from the source processor viewpoint, this is the most general type of communication, and the main problem here is that of determining which path(s) should be used to deliver source message to all its destinations. This path selection process is commonly referred to as *routing*. Since, in a DMM, there are generally many paths joining pairs of processors, routing should be done according to some criteria.

In this paper, we study routing criteria for one-to-k communication in DMMs. The two routing parameters considered are *time* and *traffic*. Parameter *time* is measured in *time steps*, where a *time step* is the actual time needed to send a unit of information (message) from a node to one of its neighboring nodes. This time is assumed to be constant for all pairs of neighboring nodes. Furthermore, we will assume that any node can inform any number of its neighboring nodes during a time step. This is contrary to the assumption made in the traditional study of system level communication in networks, by which any node could inform no more than one neighboring node during a time step [ScWu84]. Our assumption is, however, more compatible with the current technology [SaSc85]. Parameter *traffic* is quantified by the number of messages traversed in the communication links which are used to deliver the source message to its destination(s). Note that the number of time steps required to inform a destination is at least equal to the *distance* (*i.e.*, length of a shortest path) from the source to that destination. Also, the number of destinations gives a lower bound for the amount of traffic required to complete communication. Clearly, it is desirable to develop a routing scheme that completes communication while minimizing both time and traffic. It can be shown, however, that these parameters, in general, are not totally independent and thus an ordering should be used for their minimizations. We proposed that the parameter time be minimized first, that is, each destination node should receive the source message in minimum number of time steps which, of course, is equal to the distance of that node from the source. This assumption implies that the time required to receive the source message is independent of the nature and number of destinations, a phenomenon which is highly desirable in a DMM environment. Subject to this assumption, parameter traffic should be then minimized. Observe that minimizing traffic guarantees that no duplicated message will be received by any node involved in the communication.

2. PROBLEM FORMULATION AND MAIN RESULTS

Using the above routing criterion, the one-to-k communication problem in a DMM can be modeled as a graph theoretic problem as follows. Let graph $G(V,E)$ with *node set* $V(G)$ and *edge set* $E(G)$, represent the

---

[†] This research was supported in part by the DARPA ACMP Project.

topology of a a DMM. For a pair of nodes $u,v \in V(G)$, $d_G(u-v)$ represents the length (in number of edges) of a shortest path joining u and v in G. Let $M=\{u_0,u_1,\cdots,u_k\}$ be a subset of V. Node u_0 corresponds to the source node, and nodes $u_1,u_2,\cdots,u_k$ correspond to k destination nodes in a one-to-k communication. The one-to-k communication problem described above is the problem of finding a subtree $T(V,E)$ of $G(V,E)$, called *optimal communication tree* (OCT), such that

(a) $M \subseteq V(T) \subseteq V(G)$, $E(T) \subseteq E(G)$,
(b) $d_T(u_0-u_i) = d_G(u_0-u_i)$, for $1 \leq i \leq k$, and
(c) $|V(T)|$ is as small as possible.

It should be noted that if we drop condition (b) above, the above problem becomes the well-known *Steiner tree* problem [GaJo79]. Thus, *Steiner tree* problem can be used to model the situation in which the only concern is to minimize traffic.

In this paper, we first prove the following NP-hardness result using a transformation from *one-in-three-satisfiability* problem [GaJo79].

Theorem: It is NP-hard to find an OCT even if graph G is *bipartite.* □

The proof can be found in the full paper. Also, the proof of Theorem 3 in [Haki71] can be used to give another proof to the above theorem. We then present the following theorem, proof of which can be found in the full paper.

Theorem: The problem of finding an OCT can be formulated as a 0–1 *programming* problem with linear constraints. □

The above theorem is useful in two ways. First, the existing techniques for solving 0-1 programming problems can be used to generate an OCT for a given graph provided that the problem size permits. In practice, it has been observed that many 0-1 programming problems can be solved in a reasonable time. Second, as in the case of any NP-hard problem, efficient methods of finding suboptimal solutions should be explored. By formulating the OCT problem as a 0-1 programming problem, one can use the known heuristics for 0-1 programming and generate subtrees which closely approximate an OCT [Papa81, Dobs82].

Finally, we have under investigation the problem of finding an OCT when G is the n–cube graph. Currently, there are a number of DMMs whose underlying topology is n–cube, and in all such systems one-to-k communication is done by applying basically k one-to-one communications, which generally results in generating much more than required traffic [LaEN87, HMSC86, Seit85].

3. REFERENCES

[Dobs82] Dobson G., "Worst-cast Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data," *Mathematics of Operations Research*, Vol. 7, No. 4, pp. 515-531, Nov. 1982.

[Foxg83] Fox, G. C., "The Impact of Specialized Processors in Elementary Particles Physics," *Conf. on Scientific Calculation with Ensemble Computers*, Pauda, Italy, 1983.

[GaJo79] Garey, M.R. and D.S. Johnson, *Computer and Intractability*, W.H. Freeman and Company, San Francisco, 1979.

[GrRe86] D.C. Grunwald and D.A. Reed, "Benchmarking Hypercube Hardware and Software," *Technical Report*, UIUCDCS-R-86-1303, Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[Haki71] S.L. Hakimi, "Steiner's Problem in Graphs and Its Implications," *NETWORKS*, Vol. 1, pp. 113-133, 1971.

[HMSC86] J.P. Hayes, T.N. Mudge, Q.F. Stout, S. Colley, S. and J. Palmer, "Architecture of a Hypercube Supercomputer," *Proc. of 1986 Int'l Conf. on Parallel Processing*, August 1986, pp. 653-660.

[LaEN87] Lan, Y. A. Esfahanian, and L.M. Ni, "Multicast in Hypercube Multiprocessors," submitted to *J. of Parallel and Distributed Computing*, July 1987.

[Papa81] Papadimitrio C.H., "On the Complexity of Integer Programming," *J. Assoc. Comput. Mach.*, Vol. 31, pp. 392-400, 1981.

[SaSc85] Saad, Y. and Schultz, M. "Data communication in hypercubes," *Res. Rep. 428*, Dep. Comput. Sci., Yale Univ. 1985.

[ScWu84] Scheuermann P. and G. Wu, "Heuristic Algorithms for Broadcasting in Point-to-Point Computer Networks," *IEEE Transac. on Compt.*, Vol. 33, pp. 804-812, Sept. 1984.

[Seit85] Seitz, C. "The COSMIC Cube," *Communications of the ACM*, Vol. 28, No. 1, Jan. 1985, pp. 22-33.

ON SYNCHRONIC-DISTANCE AND B-FAIR RELATIONS FOR PETRI NETS

MANUEL SILVA
Depto. Ing. Electrica e Informatica
Poeta Maria Zambrano 50 (ACTUR)
50.015 Zaragoza, Spain

TADAO MURATA
EECS Department
Univ. of Illinois at Chicago
Illinois 60680, USA

ABSTRACT

Many discrete event systems such as flexible manufacturing systems can be modeled and analyzed by a graphical and mathematical tool known as Petri nets. Synchronic-distance and B-fair relations are two interesting equivalence relations for the development of a synchrony theory for Petri nets. This paper addresses the problem of determining Petri net subclasses for which both relations coincide. Also, for a didactic purpose, we present a "smallest" live and 1-bounded net for which the two relations do not coincide.

1. INTRODUCTION

Many discrete event systems such as flexible manufacturing systems can be modeled and analyzed by a graphical and mathematical tool called Petri nets [1]. Synchrony theory is a branch of General Net Theory devoted to the study of transition firing dependences. In [2], four basic functions are considered for the synchronic study of place/transition nets (Petri nets). Synchronic lead and deviation bound allow a quantification of dependences of a subset of transitions with respect to another subset. They are non-symmetric functions. Synchronic distance and fairness bound are symmetric functions, which are useful for quantifying synchronic interdependences between two subsets of transitions. A synchronic relation exists if there is any bounded synchronic function for two subsets of transitions. Synchronic relations characterize qualitatively the existence of a synchronic dependence or interdependence between two subsets of transitions. In this paper, we address some questions related to the two symmetric synchronic relations: distance synchronization and bounded fairness. In the sequel, due to the space limitation, it is assumed that the reader is familiar with the basics of Petri nets (see, for example, [3, 4, 5]).

Let $M_1 \ [\sigma\!> M_2$ represent that the application of the transition sequence σ transforms marking M_1 into M_2. $\overline{\sigma}$ represents σ's characteristic vector, where the i-th component, $\overline{\sigma}$ (i), is the number of occurrences of transition t_i in σ. We can write the following matrix equation (state equation):

$$M_2 = M_1 + C \cdot \overline{\sigma}$$

where matrix C is such that c_{ij} = Post (p_i,t_j)- Pre (p_i,t_j). Matrix C expresses the flow of tokens in N. It is called the token flow matrix, or incidence matrix (when N contains no self-loops).

A Petri net is said to be ordinary if the weight associated with each arc is one. A Free-Choice Net (FC) is an ordinary Petri net such that conflict occurs only when one place is the input to two or more transitions. In other words, conflicts and synchronization are not mixed. A Simple Net (SN) is an ordinary Petri net such that each transition has at most one input place which is shared with another transition. FC nets are a subclass of simple-nets. A Superposed Automata net (SA) is an ordinary Petri net obtained by superposing (merging) transitions belonging to different non-deterministic sequential automata (1-bounded state machines). The superposed transitions (e.g., t_2 and t_3 in figure 1b) represent the interaction between different automata.

This work was supported in part by the U.S.-Spain Joint Committee for Scientific and Technological Cooperation Grant CCB-8409024.

2. FOUR EQUIVALENCE RELATIONS

Definition 2.1: BF and SBF Relations [7, 8]:

(1) Two transitions, t_1 and t_2, in a net <N, M_o> are said to be in a bounded-fair (or B-fair) relation (BF) if there exists a positive integer k such that for any reachable marking M, and any firing sequence σ from M, $\overline{\sigma}(t_1) = 0$ implies $\overline{\sigma}(t_2) \leq k$, and $\overline{\sigma}(t_2) = 0$ implies $\overline{\sigma}(t_1) \leq k$.

(2) Two transitions, t_1 and t_2, in N are said to be in a structural B-fair relation (SBF) if they are in a B-fair relation for any initial marking, M_o.

■

Definition 2.2: SD and SSD Relations [2, 9]:

(1) Two transitions, t_1 and t_2, in a net <N, M_o> are said to be in a synchronic distance relation (SD) if there exists two non-negative integers (weights) W_1 and W_2 and a positive integer q such that for any reachable marking M, and any firing sequence or subsequence:

$$|W_1 \cdot \overline{\sigma}(t_1) - W_2 \cdot \overline{\sigma}(t_2)| \leq q.$$

(2) Two transitions, t_1 and t_2, in N are said to be in a structural synchronic distance relation (SSD) if they are in a synchronic distance relation for any initial marking, M_o.

■

The following are graphical interpretations of the SD- and BF- relations. Two transitions t_1 and t_2 are in a SD-relation if for any firable sequence or subsequence the projection of t_1 and t_2 is within two parallel straight lines with slope W_1/W_2 (see figure 2). In figure 1b all firing sequences or subsequences are projected inside a sector. If the slopes of both limiting straight lines are not zero nor infinite, transitions t_1 and t_2 are in a B-fair (BF) relation. For the net in figure 3, t_1 and t_3 are not in a BF relation because t_1 can be fired infinitely often without firing t_3 (the slope of one of the straight lines that defines the sector is zero).

From the above definitions and comments, it is not difficult to see the following.

Theorem 2.1: Behavioral Perspective:

(1) $(t_1,t_2) \in SD \Rightarrow (t_1,t_2) \in BF$, and the converse is not true.

(2) SD and BF are equivalence relations.

■

Theorem 2.2: Structural Perspective:

(1) $(t_1,t_2) \in SSD \Rightarrow (t_1,t_2) \in SBF$, and the converse is not true.

(2) SSD and SBF are equivalence relations.

■

From Definitions 2.1 and 2.2, it is obvious that:

(1) $(t_1,t_2) \in SBF \Rightarrow (t_1,t_2) \in BF$.

(2) $(t_1,t_2) \in SSD \Rightarrow (t_1,t_2) \in SD$.

3. ON THE EQUALITY OF SSD AND SBF RELATIONS

Given any pair of transitions in N, the problem we address here is that of determining net subclasses for which both relations, SSD and SBF, are the same. In particular, it is shown that this is true for pairs of transitions in live and bounded free-choice nets (LBFC nets).

Theorem 3.1 [2]: Structural synchronic relations in N are preserved after removing all the structurally marking unbounded places.

■

According to the above result, our study can be centered around structurally marking bounded nets. As an example, structural synchronic relations can be studied in the net of figure 3 after removing p_3.

Let $R = (R_1\ R_2 \cdots R_g)$ be the reproduction matrix whose columns are the marking reproduction vectors (i.e., minimal support non-negative T-invariants, $R_i \in N^m$, $R_i \neq 0$ and $C \cdot R_i = 0$).

Reproduction Matrix R can be partitioned into row vectors as follows:

$$R = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_m \end{bmatrix}, \text{ where } m = |T|$$

Let $||V||$ be the support of vector V.

Theorem 3.2 [2, 8]: Let N be a structurally bounded net:

1) t_i and t_j are in a SSD-relation iff there exist $w_i, w_j \in \mathbb{N}$ such that $w_i\, r_i = w_j\, r_j$.

2) t_i and t_j are in a SBF-relation iff $||r_i|| = ||r_j||$.

■

According to the above theorem, the following result characterizes the reproduction matrix R so that SBF and SSD for any pair of transitions are the same relation.

Theorem 3.3: Let N be a structurally marking bounded net. For any pair of transitions, $(t_i, t_j) \in T$, we have:

$(t_i, t_j) \in SBF \Longleftrightarrow (t_i, t_j) \in SSD$ iff when $||\, r_k|| = ||\, r_l\, ||$, $\exists\ \lambda_k, \lambda_l \in \mathbb{N}^+$ such that $\lambda_l\, r_k = \lambda_k\, r_l$, i.e., r_k and r_l are colinear.

■

Proof. (<=) Always holds because colinearity between r_i and r_j implies support equality (see also Theorem 2.2). (=>) The condition in Theorem 3.3 allows us to write that $||\, r_i\, || = ||\, r_j\, ||$ implies colinearity between r_i and r_j. Thus the result follows from Theorem 3.2.

■

Corollary 3.1: Let N be a structurally marking bounded net. For any pair of transitions, $(t_i, t_j) \in T$, we have

$$(t_i, t_j) \in SBF \Longleftrightarrow (t_i, t_j) \in SSD$$

iff $\forall\ t_k \in T$, $r_k = \lambda_k \cdot V_k$, where $\lambda_k \in \mathbb{N}^+$ and V_k is a (0, 1)-vector.

■

Proof. In this case, we have

$$||r_i||=||r_j|| \Rightarrow V_i = V_j \Rightarrow \lambda_j r_i = \lambda_i r_j \Rightarrow (t_i, t_j) \in SSD.$$

■

Observe that the condition in the above corollary is particularly true when R is a (0, 1)-matrix. Following the remark on the above corollary, it is easy to state the following result.

Theorem 3.4: For any pair of transitions t_i and t_j in a state machine or a marked graph, we have:

$$(t_i, t_j) \in SBF \Leftrightarrow (t_i, t_j) \in SSD$$

■

Proof. It follows from the fact that R is a (0, 1)-matrix for any state machine and for any marked graph.

■

The next question is: How much can we generalize the above theorem? The following result gives an answer.

Theorem 3.5: For any pair of transitions in any live bounded free-choice net (LBFC net), we have SSD <=> SBF.

The above equivalence does not hold in general for:

2.1. - Non-live free-choice nets
2.2. - Simple nets, even if they are live and bounded.
2.3. - Superposed automata nets, even if they are live and bounded.

■

Proof. Theorem 3.5.1 follows from the decomposition theorem in [6] stating that any LBFC net can be decomposed into a set of strongly connected marked graphs. Then, R is a (0,1)-matrix and Corollary 3.1 holds. The negative results in Theorem 3.5.2 are illustrated by the counterexamples shown in figure 4 (non-live free-choice net) and figure 5 (simple net and superposed automata net).

■

The free-choice net condition of Theorem 3.5.1 is not necessary to have SSD <=> SBF for any pair of transitions. This is illustrated by the non-simple net shown in figure 6. Finally, notice that until now we have used only the structural properties, SSD and SBF. For behavioral synchronic relations, SD and BF, all the results except for Theorem 3.1 can be adapted with the following two modifications:

1) Any marked net $<N,M_o>$, even if not bounded, can be considered.

2) Instead of using the reproduction matrix $R = (R_i)$, we must use the matrix $\sum = (\sum_i)$, where the $\sum_i$'s are the Parikh mapping (firing count vectors) of the firing sequences associated with distinct directed circuits in the coverability graph.

The justification of this result is beyond the scope of this paper.

4. A MINIMAL PETRI NET

In this section, we consider the following problem: Find a "smallest" Petri net with the following properties (where π_1 is related only to the structure of N):

π_1) Two transitions are in a SBF-relation but they are not in a SSD-relation: $(t_1,t_2) \in$ SBF but $(t_1,t_2) \notin$ SSD.

π_2) There exists an initial marking M_o on N such that $<N,M_o>$ is live and 1-bounded.

Let N be a structurally bounded net (if it is not, according to Theorem 3.1 we can remove the non-structurally bounded places). Following Theorem 3.2, to deal with π_1 we are looking for a Petri net in which there are at least two reproduction vectors, R_a and R_b such that:

$$R = (R_a \; R_b \cdots) = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

$$||r_1|| = ||r_2|| \text{ and } \nexists\, \lambda \text{ s.t. } r_1 = \lambda \cdot r_2.$$

Theorem 4.1: The minimum number of transitions in a Petri net N that satisfies π_1 is four.

■

Proof: For R_a and R_b to be two reproduction vectors (minimal non-negative T-invariants), the supports of R_a and R_b need to be incomparable (see [5]). But it is required for π_1 that $||R_a(t_i)|| = ||R_b(t_i)|| \neq \phi$ $,\forall_i = 1,2$. Therefore, to make the supports of R_a and R_b incomparable, it is necessary to add two or more transitions in addition to t_1 and t_2. Thus $|T| \geq 4$. The net shown in figure 1b satisfies π_1 and has only 4 transitions.

■

Corollary 4.1: If π_1 is satisfied, the rank of the token flow matrix C of N must be greater than or equal to 2.

Proof. It follows from the existence of at least two different reproduction vectors, R_a and R_b, with a minimum of three non-zero elements.

■

Theorem 4.2: The minimum number of places in a Petri net that satisfies both properties π_1 and π_2 is four.

■

Proof. To satisfy π_1, N must be such that there exist two reproduction vectors, R_a and R_b. The number of non-zero elements in each must be at least three (one in r_1, the second in r_2 and the third to make R_a and R_b incomparable). As r_1 and r_2 will not be colinear, one of the elements in r_1 (or r_2) must be greater than one (say $R_a(t_1) > 1$). Therefore, the number of transition firings in the firing sequence that yields R_a must be at least four. Any live and 1-bounded Petri net has (0,1)-marking vectors that need to be incomparable [5]. With 3 places, we can obtain a reachability graph of at most 3 markings [5]. Namely, the only two solutions with 3 markings and 3 places are:

$$\text{Solution } 1 \equiv \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \text{Solution } 2 \equiv \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

But in a reachability graph with 3 markings (nodes), we cannot have any directed circuit of more than 3 arcs, i.e., there is no firing sequence corresponding to R_a. Therefore, we need at least four places.

■

The net shown in figure 1b has 4 transitions (the minimum), 4 places (the minimum) and 12 arcs. A combinational study allows us to claim that the minimal number of arcs is also 12. Additionally, we conjecture that the net in figure 1b is the only one with $|P| = 4$, $|T| = 4$ and 12 arcs.

5. Conclusion

We have discussed four equivalence relations for Petri nets: BF, SBF, SD, and SSD relations; and identified subclasses of nets for which some relations coincide. In Section 3, we have discussed the question of the equality of SSD and SBF relations. Finally in Section 4, a didactic interest led us to the proof of minimality of a Petri net having certain synchronic properties. For brevity, the BF, SBF, SD, and SSD relations in this paper are defined with respect to two transitions. However, it should be noted that they can be defined with respect to two subsets of transitions as in [2, 10].

REFERENCES

[1] For example, see papers in the two special sessions on Petri Nets and Flexible Manufacturing in Procs. of the 1987 IEEE International Conference on Robotics and Automation, IEEE Computer Society Press, March, 1987.

[2] Silva, M.: "Towards a Synchrony Theory for P/T nets", Dept. Automatica, Univ. of Zaragoza, Research Report 86-10, October, 1986, also in Concurrency and Nets, Springer-Verlag, New York, 1987, pp. 435-460.

[3] Peterson, J.L.: Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[4] Murata, T.: "Modeling and Analysis of Concurrent Systems", Handbook of Software Engineering (C.R. Vick and C.V. Ramomoorthy, eds.), Chapter 3, Van Nostram Reinhold, New York, 1984.

[5] Silva, M.: Petri Nets in Automation and Computer Engineering, Editorial AC, Madrid (English Translation to be published by Kluwer Academic Press, Hingham, MA).

[6] Thiagarajan, P., Voss, K.: "A Fresh Look at Free-Choice Nets", Information and Control, Vol. 62 (pp. 195-220), 1984.

[7] Murata, T., Wu, Z.: "Fair Relation and Modified Synchronic Distances in a Petri Net", Journal of the Franklin Institute, Vol. 320, n-2, August, 1985 (pp. 63-82).

[8] Silva, M., Murata, T.: "B-fairness and Structural B-fairness in Petri Nets Models of Concurrent Systems", Dept. Automatica, Univ. of Zaragoza, Research Report 86-08, June, 1986; also, Technical Report No. UIC-EECS-86-10, Univ. of Illinois at Chicago, June 1986.

[9] Suzuki, I., Kasami, T.: "Three Measures for Synchronic Dependence in Petri Nets", Acta Informatica 19 (pp. 325-338), 1983.

[10] D. Leu, M. Silva, J.M. Colom, and T. Murata, "Interrelationships among various concepts of fairness for Petri nets", to appear.

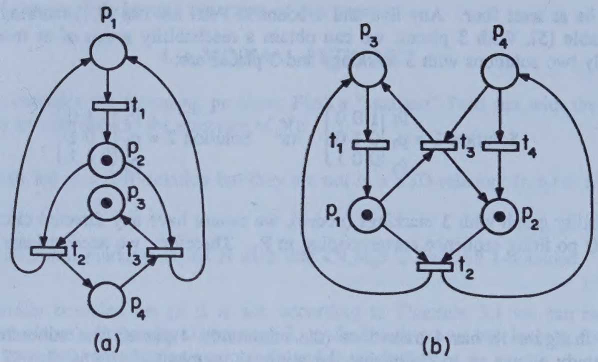

p1
t1
p2
p3
t2
t3
p4
(a)
p3
p4
t1
t3
t4
p1
p2
t2
(b)

Fig. 1. Two transitions t_1 and t_2 are in a B-fair relation in both (a) and (b), but they are not in a synchronic-distance relation in (b).

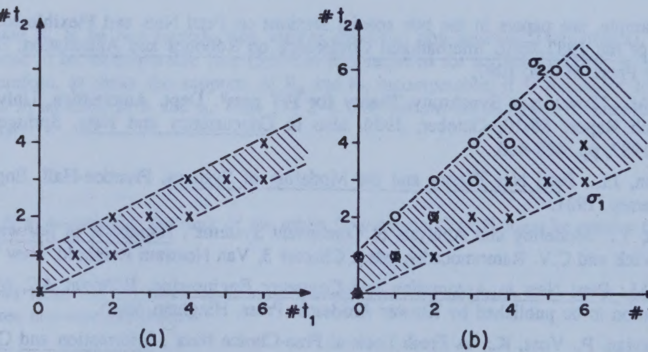


Fig. 2. The projections, t_2 vs. t_1 of all admissible sequences: (a) $\sigma = (t_2\ t_1\ t_3\ t_1)^*$ in Fig. 1(a); (b) $\sigma_1 = (t_2\ t_1\ t_3\ t_1)^*$ and $\sigma_2 = (t_2\ t_1\ t_4)^*$ in Fig. 1(b).

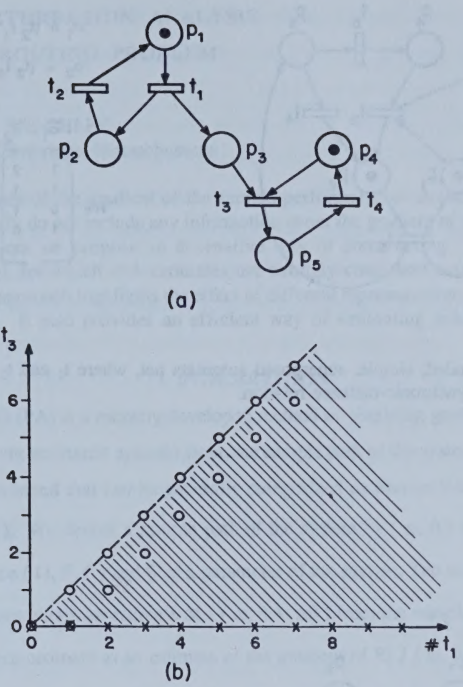
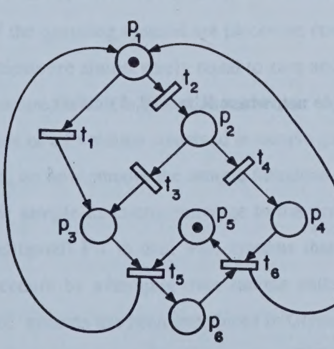

p1
t2
t1
p2
p3
p4
t3
t4
p5
(a)
#t3
6
4
2
0
0
2
4
6
8
#t1
(b)

Fig. 3. (a) A live, non-bounded, non-strongly connected marked graph (a free-choice net), and

(b) The projection, t_3 vs. t_1 of all admissible sequences.

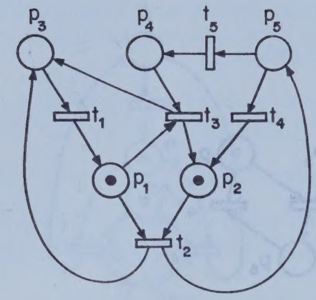

p1
t2
p2
t1
t4
t3
p3
p5
p4
t5
t6
p6

$\sigma_1 = (t_2\, t_3\, t_5\, t_2\, t_4\, t_6)^*$

$\sigma_2 = (t_1\, t_5\, t_2\, t_4\, t_6)^*$

$$R = \begin{array}{c c l} \bar{\sigma}_1 & \bar{\sigma}_2 & \\ 0 & 1 & t_1 \\ 2 & 1 & t_2 \\ 1 & 0 & t_3 \\ 1 & 1 & t_4 \\ 1 & 1 & t_5 \\ 1 & 1 & t_6 \end{array}$$

Fig. 4. A 1-bounded, non-live, free-choice net, where t_2 and t_4 are in a B-fair relation, but not in a synchronic-distance relation.

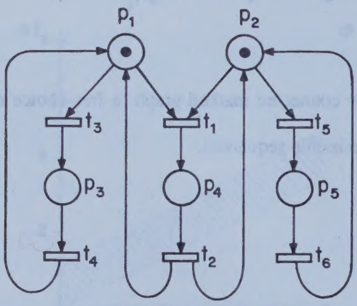


$\sigma_1 = (t_2\, t_1\, t_4)^*$

$\sigma_2 = (t_2\, t_5\, t_1\, t_3\, t_1)^*$

$$R = \begin{array}{c} \overline{\sigma_1} \quad \overline{\sigma_2} \\ \left[\begin{array}{cc} 1 & 2 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{array}$$

Fig. 5. A live, 1-bounded, simple, superposed automata net, where t_1 and t_2 are in a B-fair relation but not in a synchronic-distance relation.



$$R = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right] \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array}$$

Fig. 6. A live, 1-bounded, non-simple net, where R is a (0, 1)-matrix.

INFINITESIMAL PERTURBATION ANALYSIS
OF A MULTICLASS ROUTING PROBLEM

PIROOZ VAKILI & YU-CHI HO
Harvard University, Cambridge, Massachussetts

Abstract . The IPA estimates of the gradient of the sample performanc of stochastic systems with respect to routing probabilities typically do not include any information about the gradient of the average performance. For a multiclass routing problem we propose an alternative way of constructing sample paths of the system (alternative representation), for which IPA estimates are strongly consistent estimates of the gradient of the average performance. Our approach highligths the effect of different representation of the system on estimates of the performance gradient. It also provides an efficient way of estimating gradients in distributed routing environments.

1. INTRODUCTION

Perturbation Analysis (PA) is a recently developed method of obtaining gradient estimates of the average performance of discrete event stochastic systems from one sample path of the system. A number of different PA techniques have been developed that can be generally categorized as Fininte PA (FPA) and Infinitesimal PA (IPA) techniques (HO [6]). We denote a sample path of the system by (ω, θ) where ω is an element of the underlying probability space (Ω, F, P) and θ is a parameter of the system. The sample performance is denoted by J(ω, θ). FPA techniques obtain an estimate of J (ω, $\theta + \Delta\theta$) from the sample path (ω, θ) for a finite $\Delta\theta$ and use the finite difference estimate as an estimate of the gradient of E[J (ω, θ)]. IPA techniques calculate $\partial/\partial\theta$ J (ω, θ) and use it as the gradient estimate of the average performance.

IPA generally calculates the gradient of the sample performance function, $\partial/\partial\theta$ J (ω, θ),exactly (Suri [9]). Often the important question to ask is how much information about the gradient of the average performance, $\partial/\partial\theta$ E[J (ω, θ)], is contained in the gradient of the sample performance. It has been shown (Zazanis, Suri [10])that for a G/ G/ 1 queue under fairly general conditions the gradient of the average of the sample system times of the customers contains all the necessary information about the steady state mean system time in the system, in other words E[$\partial/\partial\theta$ J (ω, θ)] = $\partial/\partial\theta$ E[J (ω, θ)]. On the other hand, sample performance functions of finite length sample paths of the queueing systems are piecewise constant functions of routing parameters. This implies that the sample gradients are almost surely equal to zero and hence contain no information about the gradient of the performance measure (which in general is not zero).

The amount of information contained in sample gradients about gradient of average performance depends, on the one hand, on how smooth the sample functions are with respect to the parameter of interest and, on the other hand, how sample functions converge to the performance measure. Gong and Ho [5] have introduced smoothed (conditional) PA to deal with systems that have nonsmooth sample functions. They introduce a smoothing procedure by averaging over sample paths with certain charachtristics. Alternative and smooth representation of systems has been introduced in Glynn [4] for a family of random variables parametrized by a parameter of their distribution and in Glasserman [3] for a birth and death process. It is concluded in both cases, among other things, that there are alternative ways of representing the system and some of these representations are smoother than others.

It is sometimes possible to convert the problem of estimating the gradient of the average performance of the system with respect to one parameter for which the sample functions are nonsmooth to another parameter that has smooth sample functions as was shown by Ho and Cao [7] for a single class closed queueing system.

We consider the problem of routing several classes of customers that arrive to a node to a number of parallel servers (figure 1). Given conventional representation of this system, and taking delay of differnt classes of customers at each server as the performance, IPA estimates of the gradient of the sample performance contain no information about the gradient of the average performance. HO and Cao use the closed form formula of the Jackson networks to convert the problem of estimating sensitivity of throughputs with respect routing probabilities to that of sensitivity with respect to service distributions and based on that result generalize to more general networks. Closed form solutions for steady state waiting times of some special cases of the above multiclass routing problem show that the conversion method can not be applied in this case. We propose an alternative and smooth representation of this system and develop IPA algorithms that are strongly consistent. This approach, on the one hand, highlights the possibility of different representation of the stochastic systems and the fact that some of these representations are better suited than others to analyze certain aspects of these system, and on the other hand provides some insight about smoothness of the sample functions with respect to routing parameters.

With conventional representation of the systems that involve probabilistic routing small changes in the routing probabilities typically generate relatively large changes, compared to the changes in the routing probabilites (arrival or removal of a customer in a queue) that are infrequently observed. Therefore little information about the sensitivity of the system to the routing probabilities can be obtained from short observations of the system. On the other hand with the proposed alternative representation small changes in the routing probabilities generate changes of the same order on the sample path and hence sensitivity information is always available on observation periods of arbitrary length. This property is particularly desirable for optimization schemes that use sensitivity estimates from observations of the system to update the routing probabilities of the system.

The problem we consider is also of practical use. The system can serve as a node in a network of queues. Distributed routing algorithms (Gallager [2]) require that each node, based on information obtained from neighboring nodes solve an optimal routing problem at that node. To solve this problem, estimates of the gradient of delays in different queues with respect to routing probabilities are needed. Bello [1] considers customer addition and customer removal algorithms for estimating the sensitivity of the average delay in a single queue with respect to poisson arrival rate which can be used in the routing context as suggested by Segal [8] and Gallager [2]. Bello shows that the customer addition algorithm is strongly consistent for M/D/1 and customer removal is biased for M/G/1. Customer addition and customer removal algorithms are fairly inefficient algorithms in terms of their computation requirements. On the other hand IPA algorithms developed here are computationally quite efficient.

2. CONVENTIONAL REPRESENTATION

We consider a node and N parallel queues, each with infinite buffer space. M classes of customers arrive to the node and are routed to the queues. Customer of class i is routed to queue j with probability p_{ij} . W_{ij} denotes a random variable with the distribution of the steady state waiting time (not including the service time) of class i customer at queue j. Our objective is to estimate $\partial / \partial p_{kl}$ $E[W_{ij}]$ using an IPA algorithm.

To illustrate our approach we consider arrivals of two classes of customers to two queues, take $E[W_{21}]$ as the performance measure and p_{11} as the parameter of the system. The sample path of the system will be a simulation generated sample path of the system.

Conventionally, to construct the sample path of such a system, samples of several sequences of I.I.D. uniform random numbers are generated and the interarrivals, routing decisions, and servicetimes of different classes of customers are constructed using these random sequences. From the point of view of PA, the space of sequences of uniform random numbers is taken as the underlying probability space (Ω, F, P). Fixing ω in Ω, and changing θ, the parameter of the system, we obtain a parametrization of a family of sample paths characterized by being associated to the same event in the underlying probability space. The above parametrization of the sample paths also yeilds a parametrization of the sample functions defined on these sample paths. Therefore, fixing ω in Ω, we can consider $J (\omega, \theta)$ as a function of θ.

The discontinuity of the sample functions with respect to routing parameters can be best observed if we focus on the realizations of the uniform random numbers used to route the customers. In our model we focus on the realizations used to route customers of class one. Let $\omega = (\underline{u}^1 , \underline{u}^2 , ...)$ be an element of the underlying probability space Ω, $t_1 , t_2 , ... , t_n$ the arrival instances of the customers of class one to the node, and $\underline{u}^1 = (u_1 , u_2 , ... , u_n)$ the realization of the uniform random numbers used for routing of class one customers. Routing decisions are made as follows : at t_i, arrival of the ith customer, if $u_i > 1 - p_{11}$ the customer is sent to queue one , otherwise it is sent to queue two. Therefore the routing decisions can be looked at as a finite sequence of 1s and 2s. Notice that given a sequence $u_1 , u_2 , ... , u_n$, we can construct any sample of sequence of routing decisions for different routing parameters p'_{11} in [0,1] simultanuosly with that corresponding to p_{11}. Hence we have a parametrization of all the routing decision sequences corresponding to $u_1 , u_2 , ... , u_n$. More importantly for our purpose

$a = \min \{ | u_i - p_{11} | : i = 1,...,n \} > 0$ w.p.1

Therefore for all p'_{11} in the interval $[p_{11} - a , p_{11} + a]$, the routing decision sequence is identical with that of p_{11}. Conventionally, the service times of the customers are generated from a different sequence of uniform random variables and hence would not be affected by the changes in the routing parameters. This implies that for all p'_{11} in $[p_{11} - a , p_{11} + a]$ the sample path of the system given ω is identical to that of p_{11} and therefore for any sample function defined on the sample paths, the function is constant in a neighborhood of p_{11}.

The above discussion illustrates how, given the conventional representation and given ω, changes of the routing parameter of class one affects finite length sample paths of the system; For sufficiently small changes in p_{11}, no change takes place. If we change p_{11} sufficiently largely a customer of class one changes its route and either leaves or joins queue one. Since this customer has finite service time the effect of the changing of its route on the average waiting times of the customers of class two in queue one is a finite quantity w.p.1. Therefore

$W_{21}(\omega)$ is a piecewise constant function of p_{11} and it's derivative does not contain any information about derivative of $E[W_{21}(\omega)]$ with respect to p_{11}.

3. ALTERNATIVE REPRESENTATION

In this section we show how a different and smoother representation for the system can be made. Given this alternative representation, the sample gradients will no longer be identically zero.

3.1 Motivation :

Let $F_{11}(s)$ be the distribution of the servicetime of the customers of class one in queue one, and S_{11} be a random variable with distribution $F_{11}(s)$. At time t_i, arrival instant of the ith customer of class one to queue one, with probability p_{11} the customer is routed to queue one and its service requirement is generated from the service distribution $F_{11}(s)$ and with probability $(1 - p_{11})$ it is routed to queue two and no service requirement is demanded from queue one. Clearly as far as customers of class two are concerned, there would be no difference in their waiting time if the customers of class one that were routed to queue two actually were routed to queue one but their service requirement were zero. Therefore we can consider an "equivalent" system where customers of class one are routed to queue one with probability one and the random variable of their servicetime is equal to :

$S'_{11} = p_{11} . S_{11} + (1 - p_{11}) .0$ and the servicetime distribution :

$F'_{11}(s) = (1 - p_{11}) + p_{11} . F_{11}(s)$ (figure 2)

Notice that we can use the uniform random number u_i, that was used to decide the routing of the ith customer in the original system to generate the service requirement of the ith customer in the "equivalent" system:

$s'_{11} = 0$ if $u_i < (1 - p_{11})$

$s'_{11} = F^{-1}_{11}((u_i - 1 + p_{11}) / p_{11})$ if $u_i \geq (1 - p_{11})$

because :

Lemma 1: $U'_i = (U_i - 1 + p_{11}) / p_{11} \mid U_i > (1 - p_{11})$ is uniformly distributed on [0,1].

proof : Clearly $0 \leq U'_i \leq 1$, $\text{Prob}(U'_i \leq u) = \text{Prob}((U_i - 1 + p_{11}) / p_{11} \leq u \mid U_i > (1 - p_{11}))$

$= \text{Prob}((U_i - 1 + p_{11}) / p_{11} \leq u, U_i > (1 - p_{11})) / \text{Prob}(U_i > (1 - p_{11})) =$

$= \text{Prob}((1 - p_{11}) < U_i \leq p_{11} . u + (1 - p_{11})) / p_{11} = u.$

A similar argument as the one used in the above lemma can be used to show that U'_i s are independent . Therefore given the arrival instances of customers of class one we can simulate the "equivalent" system using only the uniform random numbers used to decide the routings. We use this motivating example to introduce an alternative representation.

3. 2 Alternative representation

We define $\Omega = (\underline{U}_1, \underline{U}_2, \underline{V}_1, \underline{V}_2)$ where $\underline{V}_i = (V_{i1}, V_{i2}, ...)$ i = 1, 2 are sequences of uniform random variables on [0,1] used to determine the interarrivals of class i. $\underline{U}_i = (U_{i1}, U_{i2}, ...)$ i = 1, 2 are sequences uniform random variables on [0,1] used to determine the routing and the servicetimes of class i customers. At time t_{1i},which is the arrival instant of a class one customer to the node, u_{1i},a realization of the random variable U_{1i}, is generated. If $u_{1i} \geq 1 - p_{11}$ then the customer is routed to queue one and its servicetime s_{1i} is generated by:

$s_{1i} = F^{-1}_{11}((u_{1i} - 1 + p_{11}) / p_{11})$.

If $u_{1i} < 1 - p_{11}$ then the customer is routed to queue two and its servicetime is generated by:

$s_{1i} = F^{-1}_{12}(1 - (u_{1i} /1 - p_{11}))$.

We simulate the arrival, routing and sevicetime of customers of class two similarly. From the above discussion it is clear that this is a valid representation of the system.

If p_{11} is increased by $\Delta p_{11} > 0$ to $p_{11} + \Delta p_{11}$ two types of changes could take place:

a) for all the customers of class one routed to queue one, $u_{1i} \geq 1 - p_{11}$, the service time of the customer is perturbed by an amount determined by the definition of servicetimes.

b) for $1 - (p_{11} + \Delta p_{11}) < u_{1i} < 1 - p_{11}$ the customer that originally is routed to queue two now joins queue one (figure 2). If the distribution of the servicetime of class one customers at queue one, F_{11} ,satisfy the following conditions:

i) F^{-1}_{11} is continuous and piecewise differentiable function on [0, 1]

ii) $d/dx\ (F^{-1}_{11}) < K$

then the perturbation of the servicetime of customers already in queue one will be of order of $O(\Delta p_{11})$ and the servicetimes of the new customers joining queue one will also be of order of $O(\Delta p_{11})$.

Similarly decreasing p_{11} will cause a number of customers of class one that originally are routed to queue one to be now sent to queue two. The remainig customers in queue one will have perturbed service times of order of $O(\Delta p_{11})$.

Lemma 2: For finite length sample paths $W_{21}(\omega, p_{11})$ is a continuous function of p_{11} on [0,1] w.p.1

proof: We cosider the above mentioned "equivalent" system to prove this lemma. Let $w_{2i}(\omega, p_{11})$ be the sample wait of the ith customer of class two arriving to queue one and $s'_{1j}(\omega,p_{11})$ the service time of "equivalent" jth customer of class one arriving to queue one:

$s'_{1j}(\omega,p_{11}) = s_{1j}(\omega,p_{11})$ if the jth customers is routed to queue one

$s'_{1j}(\omega,p_{11}) = 0$ if the jth customer is routed to queue two

For any p_{11} in [0,1] if we increase or decrease p_{11} by $\Delta p_{11} > 0$, the class one customers either remain in the same queue with perturbed servicetimes or switch queue. By the above discussion in any case the perturbation of the servicetime of the "equivalent" customer will be of order of $O(\Delta p_{11})$. The perturbation of "equivalent' class one customers arriving after the arrival of the ith customer of class two will have no effect on its wait and the effect of the perturbation of the servicetime of jth customer of class one arriving before the ith customer of class two on its wait can not exceed the perturbation of the servicetime itself and hence is of order of $O(\Delta p_{11})$. Let n_1 be the number of customers of class one arriving to the node. It is clear from the above argument that the effect of changing p_{11} by Δp_{11} on the wait of the ith customer of class two is at most $n_1 \cdot O(\Delta p_{11})$. This is true for all customers of class two and therefore the effect of changing p_{11} by Δp_{11} on $W_{21}(\omega, p_{11})$ is also at most $n_1 \cdot O(\Delta p_{11})$, i.e. $W_{21}(\omega, p_{11})$ is a continuous function of p_{11}.

Lemma 3: For almost all p_{11}, $d/d\, p_{11}[\, W_{21}(\omega, p_{11})]$ exists.

proof : Let $w_{2i}(\omega, p_{11})$ be as defined above. As we argued before for all p_{11} such that $1 - p_{11}$ is not one of the u_{i1}s used for routing the customers of class one in this sample path, we can find a small enough neighborhood such that no routing changes take place. Furthermore we can choose the neighborhood small enough such that no event order change on the sample path takes place as a result of the perturbation of the servicetimes of the customers of class one. This implies that the perturbation of customers of class one arriving before a class two customer in a busy period of queue one propagate fully to the waiting time of the class two customer, and no other perturbations effect the waiting time of this customer, i.e.:

$$\Delta w_{2i} = \sum_{j=1}^{n_{1i}} \Delta s_{1j}$$

n_{1i} = number of class one customers arrived before ith customer of class two in the same busy period.

Therefore $d/d\, p_{11}[\, w_{2i}\,] = \sum_{j=1}^{n_{1i}} d/d\, p_{11}\, s_{1j}$ and

$$d/d\, p_{11}[\, W_{2i}\,] = \sum_{i=1}^{N_2} \sum_{j=1}^{n_{1i}} d/d\, p_{11}\, s_{1j}$$

N_2 = number of customers of class two at queue one.

Now we are prepared to introduce an IPA algorithm.

4. IPA ALGORITHM

Here we develop an IPA algorithm to estimate the derivative of $E[W_{ij}]$ with respect to p_{kl}. This algorithm is based on the discussion of the previous sections.

Algorithm:

```
1.initialize:
  dw = 0,  dwbusy = 0,  dwsum = 0
2. At start of service of class k customer at queue j calculate $d/d\, p_{kl}\, (s_{kj})$
  dw = dw + $d/d\, p_{kl}\, (s_{kj})$
3. At start of service of class i customer at queue j
  dwbusy = dwbusy + dw
4. At end of busy period at queue j
  dwsum = dwsum + dwbusy
  dwbusy = 0
5. End of simulation
  Estimate of $d/d\, p_{kj}\, E[W_{ij}]$ = dwsum /$n_{ij}$
```

n_{ij} = number of class i customers served at queue i.

It should be noted that although in our discussion we cosidered simulation generated samples paths and alternative representation of the system, in order to use IPA algorithm it is not necessary that the samples be constructed using alternative representation. The IPA algorithm can be used for routing problems that are conventionally simulated as well as on real systems.

5. CONCLUSION

A multiclass routing problem is considered that is of practical impotance in distributed rouing environments. An alternative representation of this system is introduced that is smoother with respect to routing probabilities. An IPA algorithm is developed to estimate the gradient of average wait of different classes of customers at different queues with respect to routing probabilities. The properties of the estimator and development of algorithms to estimate the second derivative of waiting times with respect to routing probabilities are subjects of further research.

6. REFERENCES

1. M.Bello, S.M. thesis. Dep. Elec. Eng. and Comp. Sci., Massachusetts Inst. Technol., Cambridge, Sept. 1976.

2. R.G.Gallagar, A Minimum Delay Routing Algorithm Using Distributed Computation, IEEE Trans. Communication, Vol.Com-25,73-85,Jan.1977.

3. P.Glasserman, Infinitesimal Perturbation Analysis of a Birth and Death Process, Submitted to Operation Research Letter, 1987.

4. P.W.Glynn, Construction of process-Differentiable Representation for Parametric Families of Distributions, Manuscript,1986.

5. W.B.Gong and Y.C.Ho, Smoothed (Conditional) Perturbation Analysis of Discrete Event Dynnamic Systems, IEEE Trans. on Automatoc Control, to appear 1987.

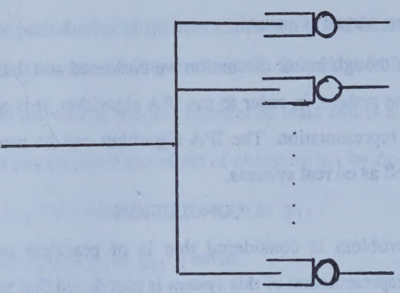
6. Y.C.Ho, Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems, IEEE trans. on Automatic Control, vol. AC-32, 563-572,July 1987.

7. Y.C.Ho and X.R.Cao, Performance Sensitivity to Routing Changes in Queueing Networks and Flexible Manufacturing Systems Using Perturbation Analysis,IEEE journal of Robotics and Automation, Vol.RA-1,165-172,1985.

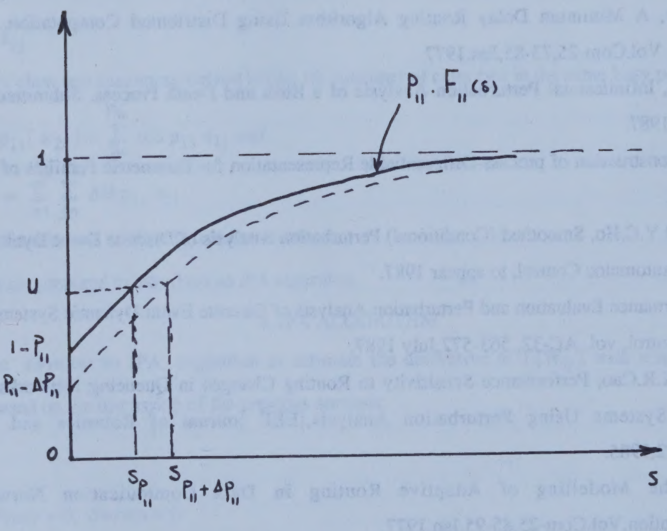
8. A.Segal, The Modelling of Adaptive Routing in Data Commuication Networks, IEEE Trans.Communication,Vol.Com-25,85-95,Jan.1977.

9. R.Suri, Infinitesimal Perturbation Analysis for General Discrete Event Systems, Journal of the ACM, to appear 1987.

10. M. Zazanis and R. Suri, Estimating Second Derivatives of Performance Measures for G/G/I Queues from a Single Sample Path, 1985, submitted to Queueing Systems.



(figure 1)



(figure 2)

AUGMENTED CHAIN ANALYSIS OF MARKOV AND SEMI-MARKOV PROCESSES [1]

STEPHEN G. STRICKLAND and CHRISTOS G. CASSANDRAS
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003

ABSTRACT

We present a new method for estimating performance sensitivities, with respect to parameters of interest, of Markov and (some) semi-Markov processes from information contained in a single nominal sample realization. Given a nominal parameter value, we use a perturbation of the parameter to define a perturbed system. We then construct what we call an *augmented chain* which in effect allows us to construct perturbed realizations from nominal ones, and hence compute sensitivities of any quantities measurable on these realizations. We show that the "observability" problem encountered in our earlier work can be overcome through an appropriate transformation of the augmented chain and present some experimental results.

1 INTRODUCTION

In performance optimization problems involving discrete event dynamic systems, analytical expressions of the performance measure (in terms of controllable parameters) generally do not exist. Gradient-based optimization methods typically employed in such cases require the *sensitivity* (or partial derivative) of the performance with respect to the parameter(s) over which the optimization is being performed. Traditionally, one resorts to simulation and uses a finite difference estimator to compute these sensitivities; this requires two simulation runs for each parameter. Recently, several new approaches have been developed, which extract information from an observed nominal state trajectory of the system in question, and directly estimate the sensitivity of the performance measure with respect to a parameter of interest. These sample path based techniques include the Likelihood Ratio Method [8],[3] and Perturbation Analysis [4],[5]-[7]. In general, these methods realize considerable computational savings as compared to the two-simulation approach since they require only a single simulation run. More importantly, since they involve only observed data, they may also be used in on-line or real-time control schemes.

This paper considers new sample path based method which has significant advantages in many situations. One advantage of this approach is that nominal system parameters need not be known. Further, the method can be applied to discrete (integer-valued) parameters (e.g. buffer capacities, routing thresholds, customer class sizes) for which the performance measures are necessarily discontinuous.

The method presumes the existence of two Markov chains whose structure is known. We assume both represent the same underlying system of interest, but differ in the value of some parameter; thus, we refer to them as the *nominal* and *perturbed* chains. Observation of a sample realization of the nominal system allows direct measurement of its sample performance. If we can *at the same time* estimate the sample performance for a perturbed realization, then we can immediately compute a finite difference sensitivity estimate. We accomplish this by using the event-driven nature of the underlying system to construct an *augmented chain* related to the nominal and perturbed chains in the following two ways: (1) The augmented chain is *stochastically similar* to the perturbed chain in that the stationary state probabilities of the perturbed chain are obtainable as the probabilities of appropriately defined aggregate states in the augmented chain. (2) The augmented chain is *observable* with respect to the nominal chain in that we can estimate the augmented chain state probabilities using information contained in a single observed nominal realization.

In previous work [1] we developed a method for constructing an augmented chain which was always stochastically similar to both the nominal and the perturbed chains; however, it was not always observable with respect to the nominal chain. We proposed a solution to this problem which involved generating additional "artificial" events to supplement those observed in the nominal sample path. This requires, however, that we know or estimate the rate parameter associated with these events. In this paper, using a more direct construction, we formalize the notion of observability and show that under some general conditions an *observable* augmented chain can always be constructed through a transformation of the initial augmented chain. We consider extensions to semi-Markov processes and present some experimental results.

---

[1] This work is supported in part by the National Science Foundation under grant ECS-8504675 and by the Rome Air Development Center under contract F302602-81-C-0169

2 DIRECT SYNTHESIS OF AUGMENTED CHAINS

In this section we extend our previous results [1] by imposing an *event structure* on the Markov chains under consideration. This allows us to obtain an augmented chain directly, as well as accomodate a more general class of Markov chains. In addition, it results in a more compact representation.

We assume that the Markov chains considered meet the following conditions:

(A1) There is a finite set of *events*; each transition defined in a chain is associated with a unique event. Moreover, all transitions associated with a given event have the same rate (i.e. the transition rate is a function of the event type alone).

(A2) For each state, there is at most one outgoing transition corresponding to each event.

Remark : We can accomodate state dependent transition rates by expanding the number of event types.

2.1 Definitions and Notation

Consider a discrete event dynamic system, represented by $\{S, E, D\}$, where S is a state space, E is a set of events which cause all possible state transitions, and D is a transition function, $D : S \times E \rightarrow S$.

Let $E^f(s)$ denote the set of events which can occur when the system is in state s; we will refer to this as the *feasible set* of events at state s. Then, given $s \in S$ and some $e \in E$, we define $D(s, e)$ as

$$D(s,e) = \begin{cases} \text{destination state when } e \text{ occurs in state } s & \text{if } e \in E^f(s) \\ 0 & \text{if } e \notin E^f(s) \end{cases}$$

If an event $e \notin E^f(s)$ occurs at state s, the event is effectively ignored and the state is unchanged. This is to be distinguished from the case of a "self-loop" transition, where $D(s, e) = s$ for some event $e \in E^f(s)$. Note that $D(s, e)$ also defines a *destination matrix* describing the system.

A Markov chain α is obtained from the discrete event system definition above by requiring that the events of each type constitute Poisson processes, and by providing an intensity function $F : E \rightarrow \Re$, characterizing each of these processes. Thus, we may write $\alpha = \{S, E, D, F\}$.

Note that we can easily obtain the infinitesimal generator of α, Q, from D and F as

$$Q(s_i, s_j) = \begin{cases} \sum\limits_{e} \delta\Big(D(s_i, e), s_j\Big) F(e) & s_i \neq s_j \\ -\sum\limits_{k \neq i} Q(s_i, s_k) & s_i = s_j \end{cases} \tag{1}$$

where $s_i, s_j \in S$, and we effectively sum the rates of all possible transitions from s_i to s_j caused by events in $E^f(s_i)$ ($\delta(\cdot, \cdot)$ is the indicator function: $\delta(x, y) = 1$ if $x = y$ and 0 otherwise).

2.2 Construction and Properties of Reduced Augmented Chains

Given the preceding notation, consider two Markov chains $\alpha_1 = \{S_1, E_1, D_1, F_1\}$ and $\alpha_2 = \{S_2, E_2, D_2, F_2\}$. For convenience, we make the following additional assumption:

(A3) α_1 and α_2 are finite, irreducible, and ergodic chains; hence, they have unique stationary state probability vectors π_1, π_2, determined by $Q_i \pi_i = 0$, $i = 1, 2$.

We then define the Reduced Augmented Chain (RAC) (the term "reduced" is used to distinguish this augmented chain from the maximal augmented chain (MAC) defined in [1]) corresponding to α_1 and α_2, as a Markov chain

$$\alpha_R = \{S_R, E_R, D_R, F_R\}$$

where

$$S_R = S_1 \times S_2$$

$$E_R = E_1 \cup E_2$$

and D_R is defined for each element $(s_i, s_j) \in S_R$ with $s_i \in S_1$ and $s_j \in S_2$ by

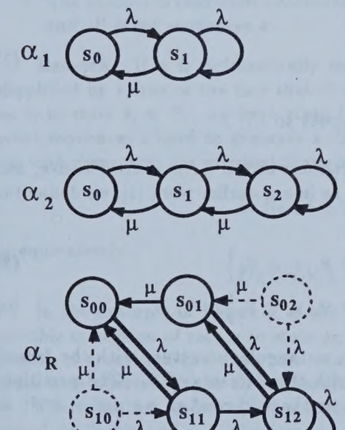
$$D_R[(s_i,s_j),e] = \begin{cases} \left(D_1(s_i,e), D_2(s_j,e)\right) & \text{if } D_1(s_i,e) \neq 0,\ D_2(s_j,e) \neq 0 \\ \left(D_1(s_i,e), s_j\right) & \text{if } D_1(s_i,e) \neq 0,\ D_2(s_j,e) = 0 \\ \left(s_i, D_2(s_j,e)\right) & \text{if } D_1(s_i,e) = 0,\ D_2(s_j,e) \neq 0 \\ 0 & \text{if } D_1(s_i,e) = 0,\ D_2(s_j,e) = 0 \end{cases} \tag{2}$$

and finally, F_R is given by

$$F_R(e) = \begin{cases} F_1(e) & \text{if } e \in E_1 \\ F_2(e) & \text{if } e \in E_2, e \notin E_1 \end{cases} \tag{3}$$

It is shown in [2] that this definition results in a RAC identical to that obtained in our earlier work [1]. A key feature of α_R is that it generally contains transient states, which can be removed since we are interested in the stationary state probabilities. A simple algorithm for constructing directly the irreducible, ergodic sub-chain of the RAC (i.e. the RAC with transient states removed), given an initial state, is also given in [2]. Hereafter, when referring to the RAC, we will assume that all transient states have been removed; thus, α_R refers to the irreducible set of states containing a given initial state.

An example of a RAC construction is shown in Figure 1, where we show the RAC which results from a nominal chain representing a homogeneous M/M/1/1 queueing system, with transition rates λ (for arrival events, a) and μ (for departure events, d), and a perturbed chain representing an M/M/1/2 system with the same transition rates. The elements S, E, D, and F of our representation are shown along with the corresponding state diagrams. Note that the construction includes two transient states, which are ignored.



$S_1=\{s_0, s_1\}$, $E_1=\{a, d\}$, $F_1(a)=\lambda$, $F_1(d)=\mu$

$$D_1 = \begin{bmatrix} s_1 & 0 \\ s_1 & s_0 \end{bmatrix}$$

$S_2=\{s_0, s_1, s_2\}$, $E_1=\{a, d\}$, $F_2(a)=\lambda$, $F_2(d)=\mu$

$$D_2 = \begin{bmatrix} s_1 & 0 \\ s_2 & s_0 \\ s_2 & s_1 \end{bmatrix}$$

$S_R=\{s_{00}, s_{01}, s_{11}, s_{12}\}$, $E_1=\{a, d\}$, $F_R(a)=\lambda$, $F_R(d)=\mu$

$$D_R = \begin{bmatrix} s_{11} & 0 \\ s_{12} & s_{00} \\ s_{12} & s_{00} \\ s_{12} & s_{01} \end{bmatrix}$$

Figure 1: *RAC For The M/M/1/1, M/M/1/2 System Pair*

In order to provide a more compact representation for D_R, let us define

$$D_R^1(s_i,e) = \begin{cases} D_1(s_i,e) & \text{if } D_1(s_i,e) \neq 0 \\ s_i & \text{otherwise} \end{cases} \tag{4}$$

$$D_R^2(s_j,e) = \begin{cases} D_2(s_j,e) & \text{if } D_2(s_j,e) \neq 0 \\ s_j & \text{otherwise} \end{cases} \tag{5}$$

which allows us to rewrite (2) as

$$D_R[(s_i,s_j),e] = \left(D_R^1(s_i,e), D_R^2(s_j,e)\right) \tag{6}$$

2.2.1 The Stochastic Similarity Property

As previously mentioned, our motivation for constructing α_R is that it contains information about the joint behavior of α_1 and α_2. In fact, if S_R is appropriately partitioned, α_R can be transformed into a Markov chain with the same stationary state probability vector as either α_1 or α_2. We refer to this property as *stochastic similarity.*

Definition : Let $\alpha = \{S, E, D, F\}$ and $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ be two Markov chains, with stationary state probability vectors π and π_0 respectively, and $\dim(S_0)=N$. The Markov chain α is said to be *stochastically similar* to α_0 with respect to P iff there exists a partition $P = \{p_i\}_{i=1,\ldots,N}$ of S such that

$$\pi(p_i) = \pi_0(s_i) \quad \text{for all } s_i \in S_0$$

Clearly, the partition P results in a Markov chain α_P. Letting Q_P and Q_0 denote the infinitesimal generators of α_P and α_0, an alternative definition is to require that $Q_P = Q_0$.

Remark : Stochastic similarity implies that the transitions between the partition sets $p_i \in P$ (as applied to S_0) constitute a realization of α_0. Thus we can explicitly extract a realization of α_0 from the observed realization of α by simply ignoring all transitions internal to any p_i.

Given the definition of stochastic similarity above, we now wish to establish the fact that the RAC, α_R, defined above, is indeed stochastically similar to both α_1 and α_2. Thus, we will show that elements of S_R can be aggregated in ways that allow us to express the stationary state probabilities of α_1 and α_2 in terms of such "aggregate" or "composite" states. The following lemma identifies the fundamental property of α_R which makes this possible.

Lemma 1: Let $\alpha = \{S, E, D, F\}$ and $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ be two Markov chains, with infinitesimal generators Q and Q_0 respectively, and $\dim(S_0)=N$. If there exists a partition of S given by

$$P = \{p_i,\ i = 1, \ldots, N\}$$

such that for all $a \in p_i$ and $j \neq i$,

$$\sum_{b \in p_j} Q(a, b) = Q_0(s_i, s_j) \tag{7}$$

where $s_i, s_j \in S_0$, then α is stochastically similar to α_0 with respect to P.

Proof : (see [2])

We shall now use this lemma to construct appropriate partitions of the RAC defined above, and establish stochastic similarity properties with α_1 and α_2. First, as in our earlier work [1], we define the following partitions of α_R:

$$P_r = \Big\{r_i : (s_k, s_j) \in r_i \text{ iff } s_k = s_i \in S_1, s_j \in S_2\Big\} \tag{8}$$

$$P_c = \Big\{c_i : (s_k, s_j) \in c_i \text{ iff } s_j = s_i \in S_2, s_k \in S_1\Big\} \tag{9}$$

The definition of S_R as the cartesian product $S_1 \times S_2$ gives it a rectangular structure with the "rows" associated with elements of S_1 and the "columns" associated with the elements of S_2. The partitions above simply formalize this fact. Thus, we will refer to $r_i \in P_r$ as the "i^{th} row" of α_R and $c_i \in P_c$ as the "i^{th} column" of α_R.

In the following result, we show that the partitions P_r and P_c allow us to establish that α_R is stochastically similar to α_1 and α_2, respectively.

Theorem 1: Let S_R be partitioned through P_r and P_c. Then, α_R is stochastically similar to α_1 with respect to P_r, and to α_2 with respect to P_c, i.e.

$$\pi_R(r_i) = \pi_1(s_i) \quad \text{for all } s_i \in S_1 \tag{10}$$

$$\pi_R(c_j) = \pi_2(s_j) \quad \text{for all } s_j \in S_2 \tag{11}$$

Proof : (using Lemma 1; see [2]).

The key implication is that a single realization of α_R provides the same information as two distinct realizations, one of α_1 and the other of α_2. Thus, we can use such a realization to obtain information regarding the behavior of both α_1 and α_2. Suppose that α_1 represents a Markov chain model of a *nominal* system, and let α_2 be identical to α_1 except for some specified parameter perturbation. Thus, α_2 corresponds to a *perturbed* system. In this context, a realization of α_R provides sufficent information to estimate performance sensitivities of the nominal system with respect to the perturbed parameter. Neither α_1 nor α_2 need be observed.

2.2.2 The Observability Property

As already mentioned, the stochastic similarity property of the RAC, α_R, presented in Theorem 1, is of interest if sample paths of α_R can be conveniently constructed, given sample paths of α_1 (the directly observable Markov chain). This is indeed possible under an *observability* condition, which we shall formalize in this section.

Consider a Markov chain $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ and let $s_0 \in S_0$ be a specified ***initial state***. Then, for any event sequence $\mathbf{e} = \{e_0, e_1, \ldots\}$, with $e_i \in E_0$ for all i, there is a corresponding state sequence $\mathbf{s} = \{s_0, s_1, \ldots\}$, with $s_i \in S_0$ for all i. Thus, $(s_0, \mathbf{e})$ describes a stochastic realization of α_0 in terms of the sequence of states visited. Note that $s_i, i = 1, 2, \ldots$ represents the state of the system *just after* event e_i occurs. In case $e_i \notin E^f(s)$, we have $s_i = s_{i-1}$ and no actual transition takes place.

Now suppose we consider a second Markov chain $\alpha = \{S, E, D, F\}$, and specify an initial state $t_0 \in S$. Given *the same* event sequence $\mathbf{e}$, we may generate a state sequence $\mathbf{t} = \{t_0, t_1, \ldots\}$ with $t_i \in S$ for all i. We define the notion of observability in terms of the relationship between $E^f(s_i)$, the feasible set of events causing transitions when α_0 is in state s_i, and $E^f(t_i)$, the corresponding feasible set of events for $t_i \in S$.

Definition : Let $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ and $\alpha = \{S, E, D, F\}$ be two Markov chains with specified initial states $s_0 \in S_0$ and $t_0 \in S$ respectively. Let $\mathbf{e} = \{e_i\}$, $i = 0, 1, \ldots$ be any event sequence with $e_i \in E_0$ for all i, and $\mathbf{s} = \{s_i\}$, $\mathbf{t} = \{t_i\}$ the corresponding state sequences for α_0, α. Then,

1. An event $e \in E$ (or the corresponding state transition from t_i to t_{i+1}) is said to be *observable* with respect to α_0 iff
$$e \in E^f(t_i) \Rightarrow e \in E^f(s_i) \text{ and } F(e) = F_0(e) \tag{12}$$
for any $i = 0, 1, \ldots$

2. The chain α is said to be *observable* with respect to α_0 iff $e \in E^f(t_i)$ is observable for all $i = 0, 1, \ldots$ and all event sequences $\mathbf{e}$

Remark : If α is stochastically similar to α_0 with respect to some partition P, the definition is simplified by virtue of the fact that P constrains the sequence $\mathbf{t}$ in terms of $\mathbf{s}$. Thus, if $\alpha = \alpha_R$, when α_0 is in state $s_i \in S_0$, α_R is in state $(s_i, s_j) \in S_R$ (for some s_j) by construction. This is true for all event sequences $\mathbf{e}$ used to generate $\mathbf{s}$. Therefore, in view of (2) and (3), the observability condition for α_R with respect to the nominal chain α_1 becomes

$$E^f[(s_i, s_j)] \subseteq E^f(s_i) \text{ for all } s_i \in S_1, (s_i, s_j) \in S_R \tag{13}$$

or equivalently,

$$D_R[(s_i, s_j), e] \neq 0 \Rightarrow D_1(s_i, e) \neq 0 \text{ for all } e \in E_R \tag{14}$$

In the example of Figure 1, if the M/M/1/1 chain is observed, then RAC state s_{01} has an unobservable transition of rate μ to state s_{00}, since the nominal state corresponding to s_{00} ($s_0 \in S_0$) has no feasible μ transition (no departure can occur when the system is empty). Note that if the M/M/1/2 chain were observed, then all events/transitions would be observable.

The definition above specifies sufficient conditions for constructing a realization of α from an observed realization of α_0. The feasible set of a state $s \in S$, along with $F(e)$ for all $e \in E^f(s)$, define the parameter of the holding time distribution (which is necessarily exponential) of the state, as well as the distribution of the type of the next event. Thus we can view observability as a condition which guarantees that the sequence of holding times and events which are observed in the realization of α_1 have the correct distributions to be used in constructing a realization of α (see [2]).

3 FULLY OBSERVABLE REDUCED AUGMENTED CHAINS

As stated previously, our goal is the construction of a RAC, α_R, which is both observable with respect to α_1 and stochastically similar to α_2. While α_R, as defined in Section 2.2, is always stochastically similar to α_2, it will often not be observable with respect to α_1 due to the constraining nature of the observability conditions. In this section, we show that, subject to some general conditions, we can transform α_R to a new RAC, α'_R, which is fully observable with respect to α_1, and retains a modified form of stochastic similarity, which we shall call *ξ-similarity*. In brief, we decompose the states of the RAC in a way that allows us to both eliminate the unobservable transitions and express the perturbed state probabilities in terms of aggregate states created by the decomposition.

3.1 ξ-Similarity

We begin by extending the definition of stochastic similarity given in section 2.2.1.

Definition : Let $\alpha = \{S, E, D, F\}$ and $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ be two Markov chains with stationary state probability vectors π and π_0 respectively, and $\dim(S_0)=N$. The Markov chain α is said to be *ξ-similar* to α_0 with respect to $V = \{V_i\}_{1,\ldots,N}$ iff there exists a set $V \subseteq S$ and a constant $\xi \in (0, 1]$ such that:

$$\pi(V_i) = \xi \pi_0(s_i) \quad \text{for all } s_i \in S_0$$

The next Lemma is a generalization of Lemma 1, and establishes conditions under which two chains are *ξ-similar* . In what follows, given a state space S, $s \in S$, and $A \subset S$, we shall use $Q(s, A)$ to denote $\sum_{t \in A} Q(s, t)$. We shall also denote the complement of A with respect to S by $\bar{A}$.

Lemma 2: Let $\alpha = \{S, E, D, F\}$ and $\alpha_0 = \{S_0, E_0, D_0, F_0\}$ be two Markov chains with infinitesimal generators Q and Q_0 respectively, and $\dim(S_0)=N$. Let P be a partition of S such that

$$P = V \cup W = \{V_i\}_{1,\ldots,N} \cup \{W_i\}_{1,\ldots,N}$$

and the following conditions hold for all $i = 1, \ldots, N$:

(C1) $Q(s, \overline{W}_i) = Q(s, V_i)$, for all $s \in W_i$,

(C2) $Q(s, V_i) + Q(s, W_i) = Q_0(s_k, s_i)$, for all $s \in V_k$, $k \neq i$

(C3) W_i is not an absorbing aggregate state

Then, α is *ξ-similar* to α_0 with respect to V, i.e.

$$\pi(V_i) = \xi \pi_0(s_i)$$

and ξ is given by

$$\xi = \pi_R(V) = 1 - \pi_R(W)$$

Proof : (by flow balancing around each V_i and W_i; see [2]).

Remark : Lemma 1 may be viewed as a special case of Lemma 2, where $V = S$ and $V_i = p_i$. In this case $\xi = 1$ and condition (C2) reduces to (7). Note that no assumptions are made regarding transitions originating in each W_i, except that any terminal states lying outside W_i must belong to the corresponding V_i.

In what follows, we will decompose S_R, the state space of α_R defined in section 2.2, so as to define a partition satisfying the conditions of Lemma 2, and also where all unobservable transitions originate within a W_i set. Using this partition, we define a transformation of α_R yielding a new RAC, α'_R, which is observable with respect to α_1 (nominal chain) and ξ-similar to α_2 (perturbed chain).

3.2 Decomposing the RAC: Active and Passive States

Without loss of generality, we assume that α_1 is the nominal chain (recall that we denote the nominal and perturbed chains, and the RAC by $\alpha_1, \alpha_2, \alpha_R$, respectively). Let U_i denote the set of RAC states in the composite state c_i (equivalently, the i^{th} column) defined in (9) which emit unobservable transitions, and let $R_i = c_i - V_i$.

Let us attach a binary indicator to the state of α_R which takes on two values: *active* and *passive* and is defined as follows. We assume the state is initially active at the start of the sample path. Entering any U_i while the state is active causes a switch to the passive state. The indicator remains passive until the system enters the R_i corresponding to the U_i which initiated the passive state. At this point it returns to the active state. Note that this indicator has no effect on the evolution of the original state vector. Further, entering any U_i while in the passive state has no effect on the binary indicator.

The RAC states $s \in S_R$ are thus decomposed into distinct component states $\{s_a, s_p^0, s_p^1, s_p^2, \ldots\}$ defined by $s_a = s$, given the system is in the *active* state, and $s_p^i = s$, given the system is in the *passive* state as a result of entering U_i.

Based on this decomposition, let us define an aggregate state W_i as the set of all i^{th} passive components of states $s \in S_R$, i.e.

$$W_i = \bigcup_{s \in S_R} s_p^i \quad \text{for } i = 1, 2, \ldots$$

$$\overbrace{\ldots R_iR_jR_iR_kR_j}^{active}\ \overbrace{U_i\ldots}^{passive}\ \overbrace{R_iR_lR_kR_j\ldots}^{active} \Rightarrow \overbrace{\ldots V_iV_jV_iV_kV_j}^{active}\,W_i\,\overbrace{V_iV_lV_kV_j\ldots}^{active}$$

Figure 2: *The RAC State Trajectory Decomposition*

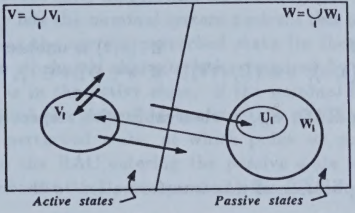


Figure 3: *Active and Passive State Space Decomposition*

In effect, the RAC enters W_i when it enters U_i while in the active state, and remains in W_i until the next visit to R_i. Thus, the RAC state trajectory can be viewed as a sequence of active segments connected by visits to a single W_i (Figure 2). Let us also use V_i to denote the set of all active components of states $s \in R_i$, and let $V = \cup_i V_i$. This defines the most important decomposition of S_R—into active and passive states (see Figure 3)—as

$$S_R = \bigcup_s s_a \ \cup \ \bigcup_{i,s} s_p^i = \bigcup_i V_i \ \cup \ \bigcup_i W_i = V \cup W \tag{15}$$

The next result establishes the fact that this decomposition of S_R satisfies the conditions of Lemma 2, and, therefore, allows α_R to be ξ-similar to α_2 with respect to the set of active states, V.

Lemma 3: Given the partition $V \cup W$ defined by (15), if $V_i \neq \emptyset$ for all i, then α_R is ξ-similar to α_2 with respect to V, i.e.

$$\pi_R(V_i) = \xi\pi_2(s_i), \quad \text{with} \quad \xi = \pi_R(V) = 1 - \pi_R(W)$$

Proof : (using Lemma 2; see [2]).

Remark : Lemma 3 establishes *ξ-similarity* between α_R and α_2 with respect to the set of active states V. The portions of the nominal trajectory during which the system occupies active states (elements of V) effectively constitute a realization of the perturbed system (this is similar to the "cut-and-paste" idea of [7]). Thus we use observations made during these portions to estimate the perturbed state probabilities. The fraction of the total observation interval which these portions constitute is given by ξ, where $0 < \xi \leq 1$. For the fastest possible convergence of our estimates, we clearly want ξ as close to 1 as possible.

Note that, by construction, all unobservable transitions originate in W (if none exist, then $W = \emptyset$ and $\xi = 1$). Thus, our next objective is to transform α_R so as to eliminate all unobservable transitions without violating (C1)-(C3).

3.3 The Observability Transformation

In this section we present our main result which states that under certain general conditions, there exists a transformation of α_R yielding a new RAC, α'_R, which is both observable with respect to α_1 and ξ-similar to α_2.

We begin by defining a state transition transformation for a Markov chain $\alpha = \{S, E, D, F\}$. Let Φ denote the set of all state transitions defined in α, i.e.

$$\Phi = \{[(s,t),e] : \ s,t \in S, \quad e \in E, \quad D(s,e) = t, \quad F(e) > 0\}$$

Then, a *state transition transformation* is defined to be a mapping:

$$T : \Phi \rightarrow (S \times S) \cup \emptyset$$

where the mapping to $\emptyset$ corresponds to removal of a transition $[(s,t),e]$. For simplicity, we shall limit ourselves to transforming transitions caused by a single event between each pair of states, and not affecting the transition rate of this event. Thus, we denote this transformation by

$$T(s,t) = (u,v) \quad \text{or} \quad T(s,t) = \emptyset$$

where $(u, v) \in (S \times S)$, and the associated event is implied.

Now, we seek a transformation T^* which, when applied to all state transitions of α_R, generates a RAC observable with respect to α_1 and also ξ-similar to α_2.

***Theorem* 2**: Let Φ_R be the set of state transitions in α_R, and T^* a transformation applied to $(s, t) \in \Phi_R$ such that

$$\begin{aligned} T^*(s,t) &= \emptyset & &\text{if } (s,t) \text{ is unobservable} \\ &= (s,v),\ v \in (R_j \cap V_i) & &\text{if } s \in W_i, t \in r_j,\ \ i,j = 1,\dots,N \end{aligned}$$

If $V_i \neq \emptyset$ for all i, the resulting RAC, α'_R, is observable with respect to α_1 and ξ-similar to α_2 with respect to V.

Proof : (see [2]).

In Figure 4 we show the application of this transformation to the example of Figure 1.

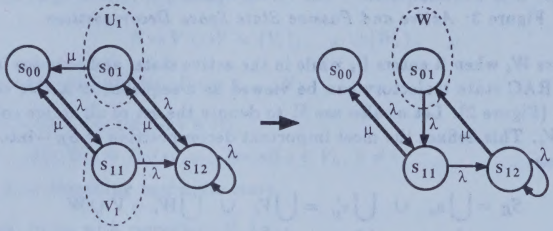


Figure 4: *Observability Transformation of the M/M/1/1, M/M/1/2 RAC*

Remark : Note that all but one of the conditions required are met either directly or indirectly by the construction of α_R. The only "real" condition is that $V_i \neq \emptyset$, for all i. This is equivalent to requiring that for each state $s \in S_2$, there exist at least one state $t \in S_1$ such that $E^f(s) \subseteq E^f(t)$. Also, note that simply removing the unobservable transitions is generally not sufficient since (as in our example—see Figure 4) it may make one or more W_i sets absorbing.

4 Extension To Semi-Markov Processes

We have investigated extensions of our method to semi-Markov processes via two directions. The first utilizes a discrete-time Markov chain imbedded in a continuous-time semi-Markov process. This only requires extending our existing methodology to discrete-time Markov chains. The second approach involves applying our continuous-time approach directly, simply relaxing the requirement that the event streams constitute Poisson processes. For certain restricted classes of semi-Markov processes, we obtain results equivalent to the pure Markovian case.

4.1 Imbedded RACs

The imbedded Markov chain approach is well known. We need only extend our augmented chain method to discrete-time Markov chains. While there are some complications due to the differing normalization conditions, this can be done. In Section 5 we apply the imbedded chain approach to the M/GI/1/K queueing system. We note in passing that the utility of the imbedded chain approach rests on the ability to relate characteristics of the imbedded chain to those of the continuous-time chain in which it is imbedded. In the case of the M/GI/1/K system, we are fortunate in that the stationary state probabilities of the (imbedded) discrete-time and continuous-time systems are identical. This is generally not the case.

4.2 Relaxation of the Markovian Assumption

In this approach, which we apply to the GI/M/1/K system in Section 5, we simply relax the Markovian requirement for one or more of the event processes (allowing them to be arbitrarily distributed). In some cases (e.g. the GI/M/1/K system) the efficacy of the method is unaffected by this relaxation. While this represents ongoing research, the basic idea can be outlined as follows.

Consider the problem of constructing a perturbed realization in a simulation environment. Given an initial state, our task can be viewed as one of constructing a "stochastically correct" sequence of state holding-times $(\tau_0, \tau_1, \ldots)$, with associated terminating events $(e_0, e_1, \ldots)$. Given a model of the system, this event sequence uniquely determines the state sequence. At each iteration i, we use e_i to determine the next state, and repeat the process. In a simulation environment, we can generate a (τ_i, e_i) pair which is "stochastically correct" by generating an exponentially distributed random number t_k for each feasible event e_k (parameterized by the event rate) and setting $\tau_{i+1} = \min_k\{t_k\}$ and $e_{i+1} =$ the associated event. The augmented chain essentially lets the nominal system perform this operation. If the current nominal state has the same feasible set as the current perturbed state (in the construction) then the observed (τ_i, e_i) pair has the appropriate stochastic characteristics required by the perturbed realization. This is what occurs when the RAC is in the active state. If the nominal feasible set does not match that of the perturbed state, we suspend the perturbed construction until the nominal system enters a state which *does* match that of the perturbed state, at which point we proceed as before. Suspension of the construction corresponds to the RAC entering the passive state via some U_i; entering a nominal state where we can restart the construction corresponds to the RAC re-entering the active state via the corresponding V_i.

In a non-Markovian environment, things are much more difficult because the (τ_i, e_i) statistics are not functions of the state alone but also of the elapsed times since the previous occurrence of each non-Markovian event. In some cases, however, we can still extract nominal (τ_i, e_i) pairs with the correct statistics. While a complete discussion is beyond the scope of this paper, this appears to be possible only when we have no more than one non-Markovian event process active at any time, when each U_i with an unobservable non-Markovian event is reachable only by transitions corresponding to that event, and where each corresponding V_i is reachable by transitions corresponding to the same event. Such a case is the M/GI/1/K system.

5 Experimental Results

In this section we provide experimental results for three variations on the single server queueing system: the M/M/1/K, M/GI/1/K, and GI/M/1/K systems (more extensive results are contained in [2]). In each case considered below, K= 2, the "GI" distribution is deterministic, and the utilization is 1 (i.e. $\lambda = \mu$). The perturbed systems represent a change in queue capacity of +1. The performance measures considered are: the utilization, U, the mean queue length, N_Q, and the mean delay (or system time), D. We applied the following variations of our augmented chain approach:

1. **URAC/K** where we use the original, unobservable RAC of Section 2 and handle unobservable transitions by generating artificial events, when required, using a random number generator parameterized by the event rate *which we assume is known.*
2. **URAC/E** which is identical to the URAC/K case except that we assume the event rate is unknown, thus we estimate it using observations made in the nominal path.
3. **IRAC** where we use an imbedded discrete-time RAC.
4. **TRAC** where we use the observability transformation of Section 4 to obtain a fully observable RAC.
5. **SIM** which is a straightforward simulation of the perturbed chain.

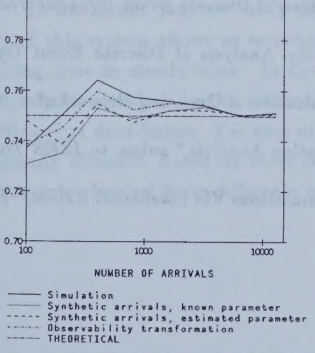


Figure 5: *M/M/1/2 Utilization Estimates*

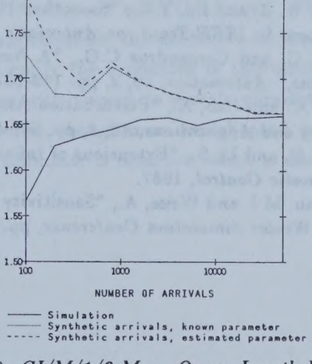


Figure 6: *GI/M/1/2 Mean-Queue-Length Estimates*

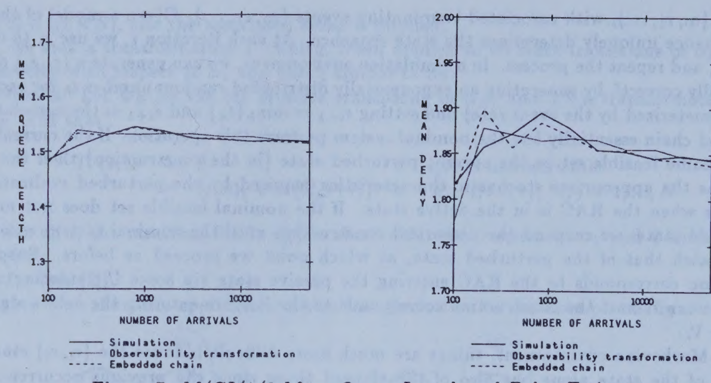


Figure 7: *M/GI/1/2 Mean-Queue-Length and Delay Estimates*

While all these techniques can be applied to Markov processes, not all can be applied to the same class of semi-Markov processes (see [2]).

For the M/M/1/2 system, we applied the URAC/K, URAC/E, TRAC, and SIM methods. In Figure 5 we show the resulting estimates of perturbed utilization as a function of the number of nominal arrivals. Also shown are the corresponding results using a straightforward simulation of the perturbed system. All curves are the average of 10 runs. Note that there is little degradation associated with estimating the artificial parameter from the nominal path as compared to a priori knowledge. Also note that the convergence of the observability-transformation method is slower than the other methods, indicating that $\xi < 1$.

We applied the and URAC/K, URAC/E, and SIM methods to the GI/M/1/2 system and the IRAC, TRAC, and SIM methods to the M/GI/1/2 system; the resulting estimates (averages of 10 runs) of perturbed mean-queue-length and delay(for the M/GI only) for these two systems are plotted versus the number of arrivals in Figures 5 and 7, respectively. The convergence of all methods is comparable.

Acknowledgement: The authors acknowledge many useful discussions with Don Towsley.

References

[1] Cassandras, C.G. and Strickland, S.G., "An Augmented Chain Approach For On-Line Sensitivity Analysis of Markov Processes," to appear in *Proc. 26th IEEE Conf. on Decision and Control*, 1987.

[2] Cassandras, C.G. and Strickland, S.G., "A General Approach For Sensitivity Analysis of Discrete Event Systems With Markovian Properties," unpublished manuscript, 1987.

[3] Glynn, P.W. and Sanders, J.L., "Monte Carlo Optimization of Stochastic Systems: Two New Approaches," *Proc. 1986 ASME Computers in Engineering Conference*, 1986.

[4] Gong, W. B. and Ho, Y.C., "Smoothed Perturbation Analysis of Discrete Event Dynamic Systems," to appear in *IEEE Trans. on Automatic Control*, 1987.

[5] Ho, Y.C. and Cassandras C.G., "A New Approach to the Analysis of Discrete Event Dynamic Systems," *Automatica*, 19, 2, pp. 149-167, 1983.

[6] Ho, Y.C. and Cao, X., "Perturbation Analysis and Optimization of Queueing Networks," *J. Optim. Theory and Applications*, 40, 4, pp. 559-582, 1983.

[7] Ho, Y.C. and Li, S., "Extensions of Infinitesimal Perturbation Analysis," subm. to *IEEE Trans. on Automatic Control*, 1987.

[8] Reiman, M.I. and Weiss, A., "Sensitivity Analysis For Simulations Via Likelihood Ratios," *Proc. of 1986 Winter Simulation Conference*, pp. 285-289.

WEAK CONVERGENCE OF SAMPLE PATH DERIVATIVES FOR THE WAITING TIME IN A SINGLE SERVER QUEUE

MICHAEL A. ZAZANIS

Dept. of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60201

ABSTRACT

A simple theoretical framework is provided to address the problem of unbiasedness of infinitesimal perturbation analysis estimates in steady state and some simple sufficient conditions are presented. These are illustrated for the case of a GI/G/1 queue and improved conditions for unbiasedness of the estimators are given.

1. Introduction

Sensitivity analysis of queueing systems is a topic that has recently received attention from a number of authors (e.g. see [GS], [HC], [RW], [RZ]). The general idea is to provide estimates for the derivative of a performance criterion of a queueing system with respect to a parameter, from the information contained in a single sample path, without the use of finite differences.

In this paper we will examine more specifically the question of unbiasedness of derivative estimates given by the infinitesimal perturbation analysis (IPA) algorithm. (For background on IPA see e.g. [HC] and [SZ]). Even though this question has already been the subject of some papers (see [C], [SZ], and references therein) the issue of unbiasedness of the *steady state* estimates has by no means been exhausted. In §, a simple theoretical framework is given which enables us to provide additional sufficient conditions insuring that IPA estimates which are unbiased for sample paths of finite length will also be unbiased in steady state.

In the second part we verify these conditions in the case of a GI/G/1 queue. The simplicity of this system allows us to obtain an explicit expression for the derivative of the expected waiting time in steady state. In fact, in order to simplify the exposition in this summary, we will restrict ourselves to the estimation of derivatives with respect to a scale parameter of the service time distribution. The case of a general parameter, though more complicated, is not conceptually different. Based on these results, a simple estimator is obtained which is shown to be superior to classical finite difference estimators.

2. Unbiased IPA estimates in steady state

Let $\{F_i\}_{i=1,2,\ldots}$ be a filtration and let $W_i(\theta)$, $i=1, 2,\ldots$, $\theta \in [a,b]$, be a sequence of random functions on (Ω, F, P) adapted to $\{F_i\}_{i=1,2,\ldots}$. Let $f_i(\theta) = E[W_i(\theta)]$. Suppose also that

A1: for all θ $W_i(\theta)$ converges weakly to a r.v. $W(\theta)$,

A2: $\sup_i E \mid W_i(\theta) \mid^p \leq M < \infty$ with $p > 1$.

Let $f(\theta) = E[W(\theta)]$. The above conditions then guarantee that $\lim_i f_i(\theta) = f(\theta)$. We will also assume that

A3: $f_i'(\theta)$, $i=1, 2,\ldots$, and $f'(\theta)$ exist for all $\theta \in [a,b]$.

The above process is an appropriate model for a large class of systems of practical interest. For instance, $W_i(\theta)$ might be the response time of the i^{th} customer arriving to a tandem network of queues whereas θ could be a parameter of the service time distribution of one of the servers in the network or a parameter of the arrival process. Assuming the system to be stable for $\theta \in [a,b]$, it is then well known that $W_i(\theta)$ converges weakly to a r.v. $W(\theta)$ representing the delay in steady state. We are interested in estimating the derivative of $E[W(\theta)]$ for sensitivity analysis or optimization purposes.

Returning to our model, let us further assume that

A4: $\dfrac{dW_i}{d\theta}$ exists w.p.1 and is F_i- measurable for all i.

(In the above example $\dfrac{dW_i}{d\theta}$ would be computed by the IPA algorithm and F_i-measurability would represent our ability to compute $\dfrac{dW_i}{d\theta}$ from the information available up to that point. For further details see [G] and [SZ]). We will also assume that

A5: $\dfrac{dW_i}{d\theta}$, $i=1, 2,\ldots$, converges weakly to some steady state r.v. $\dfrac{dW}{d\theta}$ for all $\theta \in [a,b]$.

We can now state the following

Theorem 1: Let $W_i(\theta)$, $i=1, 2,\ldots$ satisfy *A1-A5* and suppose that, addditionally, it satisfies

A6: $\dfrac{d}{d\theta}E[\dfrac{dW_i}{d\theta}] = \dfrac{d}{d\theta}E[W_i] \quad i=1, 2,\ldots$.

A7: $\dfrac{d}{d\theta}E[W_i] = f_i'(\theta)$ converges uniformly for all $\theta \in [a,b]$.

Then

$$\frac{d}{d\theta}E[W] = E[\frac{dW}{d\theta}] .$$

Proof: The proof follows immediately from a standard theorem on differentiation and uniform convergence (e.g. see [R], p.152).

Remarks: (i) In the above context, suppose that the IPA algorithm provides unbiased estimates for finite length sample paths. (This of course is assumption *A6)*. The above theorem suggests that additional conditions are required in order for IPA to also give unbiased estimates in *steady state*. A sufficient condition is *A7*.

(ii) Assumption *A7* in the above theorem can be replaced with the alternative Assumption

A7′: $f_i'(\theta)$, $i=1, 2, \ldots$, and $\lim_i f_i'(\theta)$ are continuous on $[a, b]$ and, for all i, $f_{i+1}' \geq f_i'$ on $[a, b]$.

Then from a theorem of Dini, ([R], p150), follows that $f_i'(\theta)$ converges uniformly for $\theta \in [a, b]$. Hence *A7′* implies *A7* and, for a number of systems, the former may be easier to verify than the latter, since a great deal is known about continuity and monotonicity properties of stochastic systems.

3. Sample path derivatives for the GI/G/1 queue

In this section we illustrate the above theorem for a GI/G/1 system. We will denote by C_i the i^{th} customer, by W_i the waiting time of C_i, by θX_i his service requirement, (depending on a scale parameter θ) and by A_i the interarrival time between C_i and C_{i+1}. Also, let $P(X_1 \leq x) = F(x)$ and $P(A_1 \leq x) = G(x)$. We will assume that $E[A_1] < \infty$, $E[X_1^3] < \infty$, and that the system is ergodic for all $\theta \in [a, b]$, which is equivalent to the requirement $bE[X_1] < E[A_1]$. Let us denote by W a random variable distributed according to the steady state distribution of the waiting time. Our goal is to obtain an estimator for $\frac{d}{d\theta} E[W]$ without the use of finite differences.

Let us assume that at time $t=0$ the first customer, C_1, arrives to an empty system. Also, let us designate by L_i the index of the customer who initiates the busy period in which C_i belongs. Evidently, $L_i \leq i$, the equality holding in the case where C_i initiates a busy period. For a First Come First Served system, the waiting time of C_i is given by the relationship:

$$W_i(\theta) = \max\Big(0,\ W_{i-1} + X_{i-1}\theta - A_{i-1} \Big), \quad i = 2, 3, \cdots \tag{3.1}$$

and

$$W_1(\theta) = 0 \ .$$

Under very mild assumptions it is easy to see, using for example the conditions in [Wh] that, for any given θ, $W_i(\theta)$ is differentiable with respect to θ w.p.1. (For instance, for a GI/G/1 system, the assumption that $G(\ )$, or $F(\ )$ is absolutely continuous is more than enough to guarantee this). Hence, differentiating $W_i(\theta)$ with respect to θ, we get

$$\frac{dW_i}{d\theta} = \begin{cases} 0 & \text{if } L_i = i \\[2ex] \dfrac{dW_{i-1}}{d\theta} + X_{i-1} & \text{if } L_i < i \end{cases} \tag{3.2}$$

Using (3.2) iteratively we obtain

$$\frac{dW_i}{d\theta} = \sum_{j=L_i}^{i-1} X_j \ , \tag{3.3}$$

with ill defined sums interpreted as being equal to zero.

In terms of the model of the process of the previous paragraph, we could define F_i to be σ–$\{A_1, \cdots, A_{i-1} ; X_1, \cdots X_{i-1}\}$, $i = 1, 2, \ldots$, and F_1 to be the trivial σ–field. Then it is clear that both W_i and $\frac{dW_i}{d\theta}$ are F_i--measurable.

Consider now the discrete time renewal process defined by the indices of customers who initiate busy periods, i.e. $M_0 = 0$, and $M_k = \inf\{i : L_i > M_{k-1}\}$, for $k = 1, 2, \cdots$. The increment $N_k = M_k - M_{k-1}$ is of course equal to the number of customers served in the k^{th} busy period. (Our ergodicity assumption also guarantees that $E[N_k] < \infty$).

As it is well known, W_i, $i = 1, 2, \ldots$, is a discrete time regenerative process with respect to the renewal process M_k and it is not hard to see from (3.3) that $\frac{dW_i}{d\theta}$ is also regenerative with respect to the same renewal process. It follows then from [CI] that, for all $\theta \in [a, b]$, the sequences W_i and $\frac{dW_i}{d\theta}$ converge weakly to the r.v.'s W and $\frac{dW}{d\theta}$, and furthermore that

$$E[\frac{dW}{d\theta}] = \frac{E[\sum_{i=1}^{N_1} \frac{dW_i}{d\theta}]}{E[N_1]} \ . \tag{3.4}$$

Combining (3.3) and (3.4), we get

$$E[\frac{dW}{d\theta}] = \frac{E[\sum_{i=1}^{N_1-1} \sum_{j=1}^{i} X_j]}{E[N_1]} \ . \tag{3.5}$$

Provided that one can establish that

$$E[\frac{dW}{d\theta}] = \frac{d}{d\theta} E[W] \ , \tag{3.6}$$

equation (3.5) suggests an estimator for $\frac{d}{d\theta} E[W]$ which as we show in §5 is superior to the classical estimators involving finite differences.

4. Sketch of the proof

In §3 we have argued that the sequence of waiting times W_i in a GI/G/1 queue satisfies conditions *A1*, *A4*, and *A5*. In this section we will briefly describe how to verify the remaining conditions, thus establishing (3.6). To check *A2*, choose $p = 2$ and notice that the Kiefer-Wolfovitz conditions (see [KW]) and our moment assumptions in the begining of §3 imply that $E[W^2] < \infty$. Since for all i, $E[W_i^2] < E[W^2]$, *A2* is satisfied.

For the rest of the assumptions, our task becomes much easier if, instead of the sequence of the waiting times W_i defined in (3.1), we consider the sequence

$$W_i^*(\theta) = \max\Big(0, X_1\theta - A_1, \cdots, (X_1 + ... + X_{i-1})\theta - A_1 - ... - A_{i-1}\Big), \quad i = 2, 3, ... \tag{4.1}$$

and

$$W_1^* = 0 .$$

It is easy to see that $W_i(\theta)$ and $W_i^*(\theta)$ have the same distribution for all i. Also, according to a standard result, $W^* = \lim_i W_i^*$ exists, is finite w.p.1, and is distributed according to the steady state distribution, provided that the system is ergodic. In particular,

$$E[W^*] = E[W] , \tag{4.2}$$

a result that we will use in the sequel. Next, let

$$L_i^*(\theta) = \min\{j : 0 \le j \le i-1, (X_1 + ... + X_j)\theta - A_1 - ... - A_j = W_i^*(\theta)\} . \tag{4.3}$$

Thus $L_i^*(\theta)$ is the (smallest) value of the index that maximizes the expression in (4.1). In particular notice that, when $\theta_1 < \theta_2$, $L_i^*(\theta_1) \le L_i^*(\theta_2)$. Also, define

$$L^*(\theta) = \inf\{j : (X_1 + ... + X_j)\theta - (A_1 + ... + A_j) = W^*(\theta)\} . \tag{4.4}$$

Under the conditions mentioned in the previous section, one can easily show that, for any given θ, $\frac{dW_i^*}{d\theta}$ exists w.p.1 and is equal to

$$\frac{dW_i^*}{d\theta} = \sum_{j=1}^{L_i^*} X_j . \tag{4.5}$$

The proof of (3.6) is based on the following lemmata whose proofs are straightforward and will be described briefly.

Lemma 1:

The r.v.'s $\frac{dW_i}{d\theta}$ and $\frac{dW_i^*}{d\theta}$ have the same distribution and in particular

$$E\left[\frac{dW_i}{d\theta}\right] = E\left[\frac{dW_i^*}{d\theta}\right].$$

The proof is very simple and will be omitted.

Lemma 2: $E[\frac{dW_i^*}{d\theta}] = \frac{d}{d\theta}E[W_i^*]$ for all $\theta \in [a,b]$ and $i=1, 2,\ldots$.

Proof: Let $\delta > 0$ and consider

$$\frac{1}{\delta}[W_i^*(\theta+\delta) - W_i^*(\theta)] = \frac{1}{\delta}\left[\sum_{j=1}^{L_i^*(\theta+\delta)} (\theta+\delta)X_j - A_j - \sum_{j=1}^{L_i^*(\theta)} \theta X_j - A_j\right] \tag{4.6}$$

$$= \sum_{j=1}^{L_i^*(\theta+\delta)} X_j + \frac{1}{\delta}\left[\sum_{j=L_i^*(\theta)+1}^{L_i^*(\theta+\delta)} \theta X_j - A_j\right].$$

But $\sum_{j=L_i^*(\theta)+1}^{L_i^*(\theta+\delta)} \theta X_j - A_j$ is either 0 (when $L_i^*(\theta) = L_i^*(\theta+\delta)$), or negative, (when $L_i^*(\theta) < L_i^*(\theta+\delta)$), because of definition (4.3). Hence,

$$0 \le \frac{1}{\delta}[W_i^*(\theta+\delta) - W_i^*(\theta)] \le \sum_{j=1}^{L_i^*(\theta+\delta)} X_j \le \sum_{j=1}^{L_i^*(b)} X_j,$$

and an appeal to the Dominated Convergence Theorem complets the proof.

We can now state

Lemma 3: $E[\frac{dW_{i+1}^*}{d\theta}] \ge E[\frac{dW_i^*}{d\theta}]$, $i=1, 2,\ldots$, and $\frac{d}{d\theta}E[W_i^*]$ and $\lim_i \frac{d}{d\theta}E[W_i^*]$ are continuous on $[a,b]$.

Proof: Monotonicity with respect to i is trivial. To establish the continuity of $\frac{d}{d\theta}E[W_i^*] = E[\frac{dW_i^*}{d\theta}] = E[\sum_{j=1}^{L_i^*(\theta)} X_j]$, it is sufficient to consider $E[\sum_{j=1}^{L_i^*(\theta+\delta)} X_j - \sum_{j=1}^{L_i^*(\theta)} X_j]$, $\delta>0$, and to appeal to the Monotone Convergence Theorem. The case $\delta<0$ is similar.

Finally,

$$\lim_i \frac{d}{d\theta}E[W_i^*] = \lim_i E\left[\frac{dW_i^*}{d\theta}\right] = E\left[\lim_i \frac{dW_i^*}{d\theta}\right],$$

the last equality following from the Monotone Convergence Theorem. Hence it is sufficient to show that $E[\sum_{j=1}^{L^*(\theta)} X_j]$ is continuous with respect to θ, and the proof for that is similar to the one establishing the continuity of $\frac{d}{d\theta}E[W_i^*]$.

From the above three lemmata then, and (4.2), the remaining conditions *A3, A6,* and *A7,* follow, and an appeal to Theorem 1 establishes (3.6). We can thus state

Theorem 2: For the class of GI/G/1 systems discussed in this section, IPA gives unbiased estimates of the derivative of the expected waiting time in steady state.

5. Statistical Aspects

If a sample path consisting of M busy periods is available, the following ratio estimator for $\frac{d}{d\theta}E[W]$ is suggested by (3.5)

$$\hat{W}(M) = \frac{\sum_{k=1}^{M} \sum_{i=M_{k-1}+1}^{M_k-1} \sum_{j=M_{k-1}+1}^{i} X_j}{\sum_{k=1}^{M} N_k} . \tag{5.1}$$

From the Strong Law of Large Numbers follows that $\lim_{M} \hat{W}(M) = \frac{E[\sum_{i=1}^{N_1-1} \sum_{j=1}^{i} X_j]}{E[N_1]}$ with probability 1, and hence from (3.5) that $\hat{W}(M)$ is a strongly consistent estimator of $\frac{d}{d\theta}E[W]$. It is also straightforward to show that the Mean Square Error, $E[\hat{W}(M) - \frac{d}{d\theta}E[W]]^2$, of this estimator as a function of the number of cycles observed is asymptotically $O(\frac{1}{M})$. (The details of this derivation are complicated however by the presence of ratio estimator bias). On the other hand, classical estimates requiring finite differences can be shown to have Mean Square Errors that are at best $O(\frac{1}{M^{2/3}})$ (see [FG], [ZS]). This, together with the simplicity of the estimator given in (5.1) illustrates the effectiveness of the approach presented here, whenever it is applicable.

REFERENCES

[C] Cao, X.R. (1985). "On the Sample Performance Functions of Jackson Queueing Networks", Technical Report, Division of Applied Sciences, Harvard Univ., Cambridge, MA.

[CI] Crane, M.A. and D.L. Iglehart (1975). "Simulating Stable Stochastic Systems: III. Regenerative Processes and Discrete-Event Simulations", *Operations Research, Vol. 23,* 1, 33-45.

[FG] Fox, B.L. and P.W. Glynn (1987). "Replication Schemes for Limiting Expectations", Working Paper, Dept. of Industrial Engineering, University of Wisconsin-Madison.

[G] Glynn, P.W. (1987). "Construction of Process-Differentiable Representations for Parametric Families of Distributions", Technical Report, Mathematics Research Center, Univ. of Wisconsin-Madison.

[GS] Glynn, P.W. and J.L. Sanders, (1986). "Monte Carlo optimization of stochastic systems: Two new approaches", *Proceedings of the 1986 ASME Computers in Engineering Conference.*

[HC] Ho, Y.C., and X. Cao (1983). "Perturbation Analysis and Optimization of Queueing Networks", *J. Optim. Theory Applic. 40,* 4, 559-582.

[KW] Kiefer, J. and J. Wolfowitz (1956). "On the Characterizations of the General Queueing Process with Applications to Random Walks", *Ann. Math. Stat. 27,* 147-161.

[RW] Reiman, M.I. and A. Weiss (1986). "Sensitivity Analysis of Simulations via Likelihood Ratios", *Proceedings of the 1986 Winter Simulation Conference,* J. Wilson, J. Henriksen, S. Roberts (eds.), IEEE Press, 285-289.

[SZ] Suri, R. and Zazanis, M.A. (1986). "Perturbation Analysis gives strongly consistent sensitivity estimates for the M/G/1 queue", to appear in *Management Science*

[Wh] Whitt, W. (1974). "The Continuity of Queues", *Adv. Appl. Prob. 6,* 175-183.

[ZS] Zazanis, M.A. and R. Suri (1986). "Comparison of Perturbation Analysis with Conventional Sensitivity Estimates for Stochastic Systems", Working paper # 86-123, Dept. of Industrial Engineering, University of Wisconsin-Madison.

CONSTRUCTION OF A T-BASE AND DESIGN OF A PERIODIC FIRING SEQUENCE OF A LIVE AND BOUNDED PETRI NET

Q.W.GE, T.TANIDA, N.ONO and K.ONAGA

Laboratory for Information Nets
Dept. of Circuits and Systems
Faculty of Engineering Hiroshima University
Higashi-Hiroshima-shi, 724 Japan

ABSTRACT

This paper presents a linear programming approach to i) constructing a t-base of a Petri net through use of a maximum t-invariant I_M. and ii) designing a periodic firing sequence δ together with its enabling initial token distribution M_I such that the associated firing count vector $\bar{\delta}$ is equal to a prescribed t-invariant I_0 and the initial token number $|M_I|$ is desired as small as possible.

1. INTRODUCTION

A computer control system in communication, distributed database and production-inventory often operates without deadlocks and overflows, and hence its Petri net description model must be live and bounded in transition firing. It is known[2] that such a net must possess a t-invariant I, i.e. a non-negative integer vector satisfying AI=0 for the place-transition incidence matrix $A=A^+-A^-$. The t-invariant was used to evaluate performances of scheduling system[3], database access control[4], and to prove correctness of communication protocal[5]. However few literature dealt, in the past, with computational and organizational aspects of the t-invariant, and the t-base which is defined as a minimal set of linear independent t-invariants capable of constructing any other t-invariant as a linear combination of them.

A t-invariant of a live and bounded Petri net is known[1,11] to have an interpretation of the firing count vector of a periodic firing of the transitions. Therefore the next problem to be considered is how to arrange the firing order of the transitions and how many initial tokens to be distributed among the places in order to realize this periodic firing. The token in most cases represents a resource, such as a machine, data item, cpu, disc access and synchronization mail etc, which incurs cost in its installment, and hence must be distributed into the places as economically as possible.

We present in this paper a linear programming approach to i) constructing a t-base through use of a maximum t-invariant I_M whose support, $SUP(I_M)$. is maximal in a sense of satisfying $SUP(I)\subseteq SUP(I_M)$ for any other t-invariant I. and ii) designing a periodic firing sequence δ together with its enabling initial token distribution M_I such that the associated firing count vector $\bar{\delta}$ is equal to a prescribed t-invariant I_0 and the initial token number $|M_I|$ is desired as small as possible.

In section 3. we give a linear programming formulation to calculate a single t-invariant. By modifying the objective function W and iterating the linear programming, a maximum t-invariant I_M can be obtained. Then we use the place-transition incidence matrix A_M of the subnet PN_M, which is generated by the transitions $SUP(I_M)$, to get an integer solution (not necessarily non-negative) basis $\{X_i\}$ of a linear system $A_M X=0$, and then a t-base $B=\{I_i\}$ is constructed through the formula $I_i=a_i I_M+X_i$ where initial a_i is chosen such that I_i becomes a non-negative integer vector.

In section 4, we give an optimizing methodology of constructing a multi-round periodic firing, $\delta=\delta_1\delta_2\ldots\delta_k$, and its enabling initial marking M_I, for a purpose of realizing the minimum token number $|M_I|=\sum_p M_I(p)$ under a constraint of the firing count vector $\bar{\delta}$ being equal to the prescribed t-invariant I_0. The transition firing order $J=Q_1Q_2\ldots Q_h$ and the initial marking M_I are designed through a process of reducing a Petri net PN into an acyclic net PN by means of deleting some feedback edges $F=\{e_i\}$, and of decomposing $I_0=Y_1+Y_2+\ldots+Y_k$ in such a manner that, at the i-th firing round δ_i, a transition t fires $I_i(t)$ times in succession in accordance with the ordering J. In particular, we propose an algorithm for designing a reasonably good 2-round firing by a miture of combinatorial and linear programming approaches.

2. DEFINITIONS

Here we give the basic definitions of the Petri net and define a t-base B that is different from the one defined by Sifakis[3].

[Definition 1] A Petri net is a bipartite graph expressed by $PN=\{T,P;E=E^+\cup E^-,\alpha,\beta,M_I\}$, where

(1) T: the set of transitions,

(2) P: the set of places,

(3) E^+: the set of input edges to places,
E^-: the set of output edges from places,

(4) α_e: the non-negative integer weight of edge $e=(p,t)$ in E^-,
β_e: the non-negative integer weight of edge $e=(t,p)$ in E^+,

(5) M_I: an initial token marking on the places.

[Definition 2]

(1) $A=A^+-A^-=[A^+_{pt}]-[A^-_{pt}]$ is the place-transition incidence matrix.

$$A^+_{pt}=\begin{cases}\beta_e & \text{if } e=(t,p) \text{ in } E^+\\ 0 & \text{otherwise}\end{cases}$$

$$A^-_{pt}=\begin{cases}\alpha_e & \text{if } e=(p,t) \text{ in } E^-\\ 0 & \text{otherwise}\end{cases}$$

(2) $^\bullet t$(or $t^\bullet$) is the set of input(or output) places of t, and called the pre-set(or post-set) of transition t.

(3) $^\bullet p$(or $p^\bullet$) is the set of input(or output) transitions of p, and called the pre-set(or post-set) of place p.

(4) δ is a finite sequence of transitions.

(5) $\bar{\delta}(t)$ is the number of t appearing in δ.

(6) δ is called legal on the initial marking M_I if the firing simulation of δ on M_I can be carried out all the way to the last element of δ.

(7) $M_I[\delta>M_F$ expresses that firing sequence leads to the final marking M_F from the initial marking M_I.

M_I and M_F, are related by

$$M - M_I = A\bar{\delta}.$$

If M_F is equal to M_I, then vector $I=\bar{\delta}$ is called a t-invariant. Formally a t-invariant is defined as a non-negative integer vector I satisfying $AI=0$.

[Definition 3] When a set of t-invariants satisfies the following conditions:

(1) all the element in the set are linearly independent each other,

(2) any t-invariant of the PN can be expressed as a rational combination of elements from the set,

then it is called a t-base of Petri net PN.

In the Sifakis' definition[3] of the t-base, the rational combination in the (2) is only permitted to positive coefficients but here we are permitting the coefficients to be both positive and negative. Hence, the number of the t-base elements is fewer than the one defined by Sifakis.

[Definition 4] A subnet is called a consistent component when it is generated by the set of transitions corresponding to the non-zero components of a t-invariant I. If a consistent component does not contain any consistent sub-component in it, then it is called minimal (or elementary). A consistent subnet PN_M is called maximum if any consistent component is included in it.

[Definition 5] The support of a firing count vector I, SUP(I), is defined as the set of transitions $\{t\}$ with $I(t)\neq 0$.

3. CONSTRUCTION OF A T-BASE VIA LINEAR PROGRAMMING

In this section, we deal with the problem of constructing a t-base and present a linear programming approach to it.

A maximum t-invariant I_M of a PN is a t-invariant such that $SUP(I_M)$ is equal to the transition set T_M of a maximum consistent subnet PN_M.

Suppose we are given PN_M and I_M. We are to show how to construct a t-base B from them.

Solving the following linear equation (1),

$$\begin{cases} A_M X=0 \\ A_M = \text{P-T incidence matrix of } PN_M \\ X=\text{integer vector(not necessarily non-negative)}. \end{cases} \qquad (1)$$

we get a solution basis $\{X_i \mid i=1,2,\dots,k\}$, where $k=|T_M|-\mathrm{rank}(A_M)$. Further, using I_M and X_i we compute $\{I'_i \mid i=1,\dots,k\}$ as follows:

$$\begin{cases} I'_1=a_1 I_M+X_1 \\ I'_2=a_2 I_M+X_2 \\ \cdot \\ \cdot \\ I'_k=a_k I_M+X_k \end{cases} \qquad (2)$$

where, a_i is so chosen that each I_i becomes a non-negative integer vector.

[Lemma 1] A t-base of PN_M can be selected from $\{I'_1,I'_2,\dots,I'_k,I_M\}$ by choosing k independent vectors.

(Proof) Consider a (k+1) x m matrix Z whose columns consist of $X_1,X_2,\dots,X_k,I_m$

$$Z = (X_1\ X_2\ \dots\ X_k\ I_M)^t$$

Due to the mutual independence of $X_1,X_2,\dots,X_k$, the rank of C is at least k.

Performing column operations of adding $a_i I_M$ to X_i, we obtain a matrix

$$Z' = (I'_1\ I'_2\ \dots\ I'_k\ I_M)^t$$

where rank is equal to that of C. Hence there exist k mutually independent rows. QED

[Lemma 2] Let I_i be an extension of I_i' such that $I_i(t)=I_i'(t)$ for $t \in T_M$ and $I_i(t)=0$ for others. Then a t-base of Petri net PN can be selected from $\{I_1, I_2, \ldots, I_k, I_M\}$.

(Proof) Let I be any t-invariant of PN. Because PN_M is the maximum consistent subnet of PN, the component $I(t)$, $t \notin T_M$, is always zero. Since a t-base of PN_M can be selected from $\{I_1', I_2', \ldots, I_k', I_M\}$, so is a t-base of the original PN from $\{I_1, I_2, \ldots, I_k, I_M\}$. QED

In Lemma 1 and Lemma 2, we have shown how to construct a t-base of a Petri net through maximum consistent subnet PN_M and its t-invariant I_M.

A rational non-negative solution I_0 of $AI_0=0$ can be formulated by the following linear programming LP(3) and obtained by the simplex method.

$$\text{minimize:}\quad W=\sum_{i=1}^{n+1} z_i$$

$$\text{subject to:}\quad \begin{vmatrix} a_{11} & \cdots & a_{1m} & 1 & & & \\ \vdots & & \vdots & & 1 & & \\ \vdots & & \vdots & & & \ddots & \\ a_{n1} & \cdots & a_{nm} & & & & \vdots \\ 1 & \cdots & 1 & \cdot & \cdot & \cdot & 1 \end{vmatrix} \begin{vmatrix} I_0 \\ Z \end{vmatrix} = \begin{vmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{vmatrix} \qquad (3)$$

$$Z,\ I_0 \geq 0,\ m=|T|,\ n=|P|,$$

where a_{ij} is the element of P-T incidence matrix and $z_i (i=1,2,\ldots n+1)$ is the artificial variable.

[Lemma 3] If the optimal value of the LP(3) is zero $W=0$, then a rational solution I_0 of LP(3) can be converted into a t-invariant by multiplying an appropriate integer and I_0 is a basic feasible solution.

(Proof) From the formulation LP(3), $W=0$ implies that $AI_0=0$ and $I_0 \geq 0$. Because I_0 is rational, we can convert it into an integer vector I_0. Any feasible solution of linear programming obtained by the simplex method is basic[12], hence I_0 is a basic feasible solution. QED

Let's consider how to use the linear programming formulation LP(3) to compute the maximum t-invariant. We set $I_v := I_0$ and use I_0 as an initial basic feasible solution to carry out the simplex iteration of the following linear programming LP(4), which is reformed from LP(3) by deleting the artificial variables $\{z_i \mid i=1,2,\ldots,n+1\}$ and rewriting the objective function W as following,

$$\text{minimize:}\quad W=\sum_{i=r}^{m} I(t_i)$$

$$\text{subject to:}\quad \begin{vmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \\ 1 & \cdots & 1 \end{vmatrix} \begin{vmatrix} I \end{vmatrix} = \begin{vmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{vmatrix} \qquad (4)$$

$$I \geq 0,$$

where $I_v(t_1)=I_v(t_2)=\ldots=I_v(t_{r-1})=0$ and $I_v(t_i) \neq 0 (r \leq i \leq m)$ are holding.

The following lemma gives the properties of a new rationalsolution I obtained from LP(4).

[Lemma 4] A rational solution I of LP(4) satisfies

(1) $SUP(I) \neq SUP(I_0)$, when the optimum value is less than unity $W<1$,

(2) $SUP(I)=SUP(I_0)$, when the optimum value is equal to unity $W=1$.

(Proof) (1) We have the constraints $\sum_{i=1}^{m} I(t_i)=1$ in LP(4), the optimum value $W=\sum_{i=r}^{m} I(t_i)<1$ must imply there exists some $1 \leq j \leq r-1$ for which $I(t_j) \neq 0$ holds but

$I_0(t_j)=0$. Hence $SUP(I)\neq SUP(I_0)$. when $w<1$.
(2) Similarly it is straight forward to show $SUP(I)\subseteq SUP(I_0)$ when $W=1$. Since the simplex method produces a basic feasible solution I_0 to LP(3) that is elementary in a sense that I_0 is never expressed as a linear combination of other basic solutions. say $I_0=I_1+I_2$. So $SUP(I)\subset SUP(I_0)$ is impossible and hence $SUP(I)=SUP(I_0)$ must hold. QED

Now, we update $I_V:=I_V+I$ and use the new solution I as the initial basic feasible solution to carry out the simplex iteration of LP(4). Repeating this until $W=1$, finally a maximum t-invariant $I_M:=kI_V$ can be obtained, where k is so chosen that I_M becomes a non-negative integer vector. We give the proof in the following theorem

[Theorem 1] Carry out the simplex of the LP(3) to obtian a basic feasible solution I_0, then set $I_V:=I_0$ and carry out simplex of the LP(4) to obtain another rational solution I by using I_0 as the initial basic feasible solution. Further by repeating the following:

(1) update $I_V:=I_V+I$ and $I_0:=I$,

(2) carry out the simplex of the LP(4),

until $W=1$. finally we can get a maximum t-invariant $I_M:=kI_V$, where k is so chosen that to make I_M integer.

(Proof) From the proof of Lemma 4, we can conclude that when LP(4) is repeatedly carried out until $W=1$, the following:

$SUP(I_V)\supseteq SUP(I)$, when $W=1$,

hold. Hence. the obtained $I_M:=kI_V$ is a maximum t-invariant. QED

We apply Theorem 3 to give an algorithm to compute a maximum t-invariant I_M and the maximum consistent subnet PN_M of a given Petri net PN.

<< Maximum T-invariant I_M >>

1• Set $I_V=0$.

2• Carry out the simplex iteration on LP(3) to obtain a rational solution I_0. Otherwise the Petri net does not possess any t-invariant. If $W=0$, then do $I_V:=I_V+I_0$.

3• Use I_0 as the initial basic feasible solution to carry out the simplex iteration on LP(4) to obtain a rational solution I.

4• While $W\neq 1$. do $I_V:=I_V+I$, $I_0:=I$ and repeat 3•

5• Set $I_M:=kI_V$ and PN_M as the subnet generated by $SUP(I_M)$ and stop, where k is so chosen that to make I_M integer.

[Corollary 1] The I_M and PN_M obtained by the above algorithm are a maximum t-invariant and a maximum consistent subnet.

Example 1: Apply the "Maximum T-invariant I_M" algorithm to the Petri net of Fig.1 to compute a maximum t-invariant I_M.

1) Set $I_V:=0$.

2) By carrying out simplex method on LP(3) of the PN. we get $I_0=(0.5\ 0.5\ 0)^t$ and $W=0$. Update $I_V:=I_V+I_0=(0.5\ 0.5\ 0)^t$.

3) Using I_0 as the initial basic feasible solution to carry out simplex method on LP(4), we get $I=(0\ 0.5\ 0.5)^t$ and $W=0.5$. Update $I_V:=I_V+I=(0.5\ 1\ 0.5)^t$, $I_0:=I=(0\ 0.5\ 0.5)^t$.

4) Again using I_0 to carry out simplex method on LP(4), we get $I=(0.5\ 0.5\ 0)^t$ and W=1.

5) Since W=1, the recurrence, 3° and 4°, of the algorithm terminates and $I_M:=kI_V=k(0.5\ 1\ 0.5)^t$(k is set to 2).

Therefore, a maximum t-invariant I_M of the Petri net is

$$I_M=(1\ 2\ 1)^t$$

and the maximum consistent subnet is itself.

4. DESIGN OF A PERIODIC FIRING WITH A PRESCRIBED FIRING COUNT VECTOR I_0

We have proposed a design method of a periodic firing in our previous paper[13]. In this section, we introduce our design method and give a reasonably good 2-decomposition of a prescribed firing count vector via linear programming.

Suppose we are given a firing count vector I_0 satisfying $AI_0=0$ and required to design a periodic firing δ and its initial marking M_I. It is well known[1,11] if we put A^-I_0 tokens to the places, each transition t can independently fire $I_0(t)$ times. Here, we try to design a firing sequence which is legal on an initial marking M_I with less tokens than A^-I_0.

To design a periodic firing with minimal tokens, we choose an approach of setting a linear firing order first via an acyclic graph resulted from deleting a minimal weight feedback edge set[13]. For the minimal weight feedback edge set problem, we give in the following a heuristic algorithm.

<< Acyclic Formation Algorithm >>

1° Form a set $W=\{\beta_e I_0(t) \mid e\in E^+,\ e=(t,p)\}$. Set $F=\phi$, $M_I=0$.

2° Sort W in the non-decreasing order, $Z=\{z_1,z_2,\ldots z_k\}$.

3° Take the first element of Z and check if the corresponding edge e=(t,p) is in some cycle of PN. If so, put $\beta_e I_0(t)$ tokens in place p and update $M_I(p):=M_I(p)+\beta_e I_0(t)$, $F:=F\cup\{e\}$ and $PN:=PN-\{e\}$. Otherwise do nothing.

4° Delete the first element of Z. If Z is not empty return to 3°.

5° Stop. The final net $\widetilde{PN}$ is now acyclic.

[Theorem 2][13] Let Q_1 be the set of source transitions possessing no input places of $\widetilde{PN}$ found when the source places possessing no input transitions $S_1(=S)$ are removed. Let S_2 be a set of source places, found when the source transitions Q_1 are removed. Recursively define $Q_2,S_3,Q_3,\ldots,Q_h$ until $\widetilde{PN}$ becomes empty. Then transition sequence $J=Q_1Q_2\ldots Q_h$ can be made into a firing sequence legal on M_I such that each transition t in each Q_i is independently firable $I_0(t)$ times in succession and firings of Q_i follow these of $Q_{i-1}(i=2,\ldots,h)$. The initial marking is now expressible by $M_I=A_F I_0$(less than $A^- I_0$), where A_F is the incidence matrix related to F such that it is derived from A by setting all elements A_{pt}, corresponding to edge e=(p,t) or (t,p), to zero except those e in F.

In order to further reduce initial tokens, we decompose I_0 into 2-components $I_0=Y_1+Y_2$ for the purpose of designing a 2-round firing $\delta=\delta_1\delta_2$ in one period such that transition t fires $Y_1(t)$ times in the first round δ_1 and Y_2 times in the second round δ_2.

We need the following definitions.

(a) $T_F=\{t|e=(t,p)\in F\}$=set of transitions incident with feedback edges. $\bar{T}_F=T-T_F$;

(b) $P_F=\{p|e=(t,p)\in F\}$=set of places incident with feedback edges. $\bar{P}_F=P-P_F$;

(c) A_F=place-transition incidence matrix for the feedback edge set F. Formally the pt element $A_F(p,t)$ is defined by

$$A_F(p,t)=\begin{cases} A^+(p,t), & e=(t,p)\in F, \\ 0, & \text{otherwise;} \end{cases}$$

(d) A_{PF}=place-transition incident matrix for outgoing edges from places in P_F, formally the p-th row $A_{PF}(p)$ is defined by

$$A_{PF}(p)=\begin{cases} A^-(p), & \text{if } p\in P_F, \\ 0, & \text{otherwise.} \end{cases}$$

<< Two-round Firing >>

```
1°  Initialization:
  1.1°  Label each transition $t\in T_F$ with a given positive integer $Y(t)$ less
        than $I_0(t)$, and transition $t\in \bar{T}_F$ with $Y(t)=0$;
  1.2°  Calculate $\tilde{M}_I=A_F Y$, label each place p with $L(p)=\tilde{M}_I(p)$ and set $D(p)=0$.
2°  For each sink transition t of the acyclic net $\widetilde{PN}=PN-F$, do:
  2.1°  For each place $p\in {}^\bullet t$, update $L(p)$ as $L(p):=L(p)-\alpha_e Y(t)$, where $e=(p,t)$
        is outgoing from p;
  2.2°  Update $\widetilde{PN}$ by deleting the sink transition t and its input edges.
3°  For each sink place p of the $\widetilde{PN}$, do:
  3.1°  Calculate $R=L(p)+\sum_{e=(t,p)}\beta_e Y(t)$, where $t\in {}^\bullet p$. If $R<0$, do:
    3.1.1°  If p is in $P_F$, increase label $Y(t)(\leq I_0(t))$ of t in ${}^\bullet p$ in order to
            assure $R\geq 0$. If $R<0$ remains even when all $Y(t)$s are increased up to
            the ceiling $I_0(t)$, then set $D(p)=-R$ and $R=0$.
    3.1.2°  If p is in $\bar{P}_F$, increase label $Y(t)(\leq I_0(t))$ of t in ${}^\bullet p$ in order to
            assure $R\geq 0$ (note always possible).
  3.2°  Update $L(p)$ as $L(p):=R$, and update $\widetilde{PN}$ by deleting p and its input
        edges.
4°  If $\widetilde{PN}\neq\phi$, go to 2°.
5°  Let $Y_1=Y$, $Y_2=I_0-Y$ and $M_I=\tilde{M}_I+D+\Delta$, where $\Delta=0v((A_{PF}-(A^+-A_F))Y_2-(\tilde{M}_I+D+AY_1))$.
    Stop.
```

[Theorem 3][13] Let Y_1, Y_2 and M_I be those obtained from the above algorithm. Then 2-round firing $\delta=\delta_1\delta_2$ with $\bar{\delta}_1=Y_1$ and $\bar{\delta}_2=Y_2$ is legal on M_I, and the token number $|M_I|$ is not greater than that of single-round firing.

In the above algorithm, the initial vector Y is not concretely given. In fact. the Y is related to a good 2-decomposition of $I_0=Y_1+Y_2$ for achieving a fewer initial tokens. In the following discussion, we consider a problem of how to obtain get a reasonably good Y.

The optimization of the minimal initial marking of a periodic firing can be formulated in integer programming under a fixed set of feedback edges, F.

Let $I_0=Y_1+Y_2$ be any non-negative integer decomposition of I_0. Let M_I^i be the marking for $Y_i(i=1,2)$ computed by $M_I^i=A_F Y_i$.

Starting from M_I^1 as a temporal initial marking. We try to make Y_1 firable, first by supplementing an additive marking D_1 if neccessary and then Y_2 firable next in the similar manner.

Find the minimal additive D_1 so as to assure $D_1+AY_1 \geq 0$. Minimal D_1 is given by the following closed form with "v" as the component-wise maximum operator.

$$D_1 = 0 \mathrm{v} (-AY_1).$$

Then the source transition sequence $J=Q_1Q_2\ldots Q_h$ of acyclic $\widetilde{PN}$ can be made into a firing sequence δ_1, legal to $M_I^1+D_1$, in which each t in Q_j is independently firable $Y_1(t)$ times in succession. δ_1 leads to marking M_F^1 from $M_I^1+D_1$ such that

$$M_F^1 = M_I^1+D_1+AY_1$$

holds.

Find the minimal additives G_2 so as to assure

$$M_F^1+G_2 \geq M_I^2$$

$$G_2 \geq 0$$

Then again $J=Q_1Q_2\ldots Q_h$ can be made into a firing sequence δ_2, legal to $M_F^1+G_2$, in which each t in Q_j fires independently $Y_2(t)$ times in succession. Now it is clear that a concatenated firing sequence $\delta_1\delta_2$ is legal to the initial marking $M_I^1+D_1+G_2$ and possesses the prescribed firing-count vector Y_1+Y_2.

Let $|D|$ denote the sum of the components of a vector D and let W the token economy gained by the two-round firing over the single round firing. Then our optimization problem of the token number in the initial marking M_I can be formulated as the following integer programming.

$$\begin{aligned} &\text{maximize } W=|M_I^0|-\{|M_I^1|+|D_1|+|G_2|\} \\ &\text{subject to } I_0=Y_1+Y_2 \\ &\qquad M_I^0=A_F I_0 \\ &\qquad M_I^i=A_F Y_i\ (i=1,2) \\ &\qquad D_1+AY_1 \geq 0 \\ &\qquad M_I^1+D_1+AY_1+G_2 \geq M_I^2 \\ &\qquad Y_i, D_1, G_2\text{: non-negative integer vector}(i=1,2). \end{aligned} \tag{5}$$

However. integer programming costs huge computation time and is not practical. Hence, we change the constrain "Y_i,D_1,G_2: non-negative integer vector" simply to "Y_i,D_1,G_2: non-negative vector" so as to convert (5) into a linear program. To our problem, we use some integer vectors Y around the real solution Y_1 thus obtained.

Example 2: To design a two-round periodic firing and its initial marking of the Petri net shown in Fig.2 with a t-invariant(firing count vector) $I_0=(4\ 3\ 2\ 2\ 2\ 2\ 2)^t$.

Applying "Acyclic Formation Algorithm", we get a set of feedback edges F and transition sequence J:

$$F=\{(t_4,p_3),(t_3,p_4),(t_5,p_3),(t_2,p_2)\},$$

$$J=\{t_3,t_6\}\{t_1,t_5\}\{t_2,t_4\}.$$

If the two-decomposition of I_0, obtained by "Two-round Firing" algorithm, is

$$I_0=Y_1+Y_2=(2\ 1\ 1\ 1\ 1\ 1)^t+(2\ 2\ 1\ 1\ 1\ 1)^t,$$

then the initial marking is

$$M_I=(0\ 5\ 3\ 1\ 0\ 0)^t.$$

5. CONCLUSION

A t-invariant is a non-negative integer vector I satisfying AI=0 and its computation is basically formulated as an integer programming problem. Due to the homogeneousness of AI=0, it is possible to use rational simplex iteration for computing a single t-invariant and constructing a t-base, which is defined as a minimal set of t-invariants capable of generating any other t-invariants through maximality device. Further more we have presented a methodology of designing a periodic firing sequence of the prescribed firing count vector and proposed a linear programming formalism to optimize the token number in the initial marking.

Acknowledgement: The authors wish to acknowledge that a portion of this research has been supported by Mazda Fundations' Science and Technology Grant 1986-88.

References

(1) J.L.Peterson: "Petri Net Theory and The Modeling of Systems", Prentice-Hall, (1981).

(2) C.Ramchandani: "Analysis of Asynchronous Concurrent Systems by Timed Petri nets", Ph.D. Thesis, MIT, Sept. (1973).

(3) J.Sifakis: "Structural Properties of Petri Nets", Proceedings of the Seventh Symposium on Mathematical Foundations of Computer Science 1978, Lecture Notes in Computer Science, Vol.64, Berlin: Springer-Verlag, pp.474-483, (1978).

(4) M.T.OZsu: "Modeling and Analysis of Distributed Database Concurrency Control Algorithms Using an Extended Petri Net Formalism", IEEE Trans. Software Eng., SE-11. 10. pp1225-1240, (1985).

(5) K.Garg: "An Approach to Performance Specification of Communication Protocols Using Timed Petri Nets", IEEE Trans. Software Eng., SE-11. 10. pp.1216-1225. (1985).

(6) J.Sifakis: "Use of Timed Petrinets for Performance Evaluation", in 3rd Int. Symp. Measuring, Modeling and Evaluating Comput. Syst., Beilner and Gelenbe. Eds., pp.75-95. (1977).

(7) M.Silva and T.Murata: "B-Fairness and Structural B-Fairness In Petri Net Models of Concurrent Systems", Technical Report #86-10. (1986).

(8) T.Murata: " Petri Nets. Marked Graphs. and Circuit-System Theory", IEEE Circuits and Systems Society Newsletter. Vol.11. No.3, pp.2-12.(1977).

(9) R.M.Karp and R.E.Miller: "Properties of A Model for Parallel Computations: Determinacy, Termination. Queueing", SIAM J.App.Math., Vol.14. No.6, pp.1390-1411. (1966).

(10) F.Commoner and A.W.Holt: "Marked Directed Graphs", Journal of Computer and System Sciences, Vol.5, No.5, pp.511-523, (1971).

(11) K.Hiraishi and A.Ichikawa: "Initial Marking at Which Firing Count Vector Has Firable Firing Sequences in Petri Net", The 2nd National Net Theory

Workshop (in Japanese) Osaka, Japan, pp.33-40. (1986).

(12) L.S.Srinath: "Linear Programming Principles and Applications", The Macmillan Press LTD, London and Basingstoke, (1983).

(13) K.Onaga, Q.W.Ge and N.Ono: "On Optimizing the Initial Token Distribution for a Periodic Petri Net Firing Sequence with Prescribed Firing Numbers", to be published in Trans. of the Society of Instrument and Control Engineers, Vol.23, No.10. (1987).

(14) J.A.Bondy and U.S.R.Murty: "Graph Theory Applications", The Macmillan Press LTD, London, (1976).

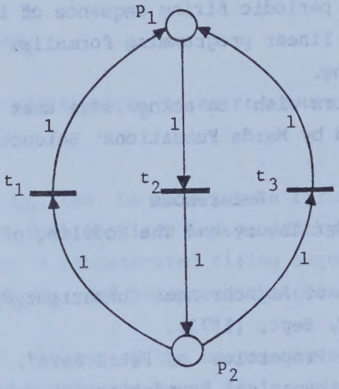

p_1
1
1
1
t_1
t_2
t_3
1
1
1
p_2

Fig.1 An example Petri net.

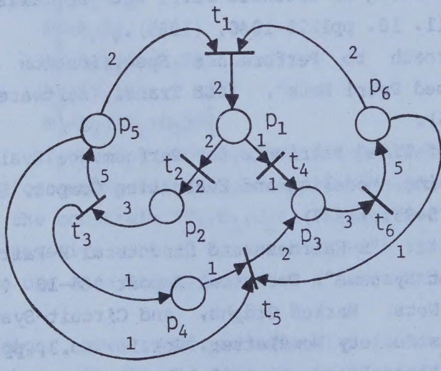

t_1
2
2
2
p_6
p_5
p_1
2
1
t_2
t_4
2
1
5
5
3
t_3
p_2
p_3
3
t_6
2
1
1
1
t_5
p_4
1

Fig.2 A Petri net possessing a t-invariant

$I_0=(4\ 3\ 2\ 2\ 2\ 2)^t$.

T-Complement Petri nets

R.S. Sreenivas
Department of Electrical and Computer Engineering
and the Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

SUMMARY

Petri nets are a cogent tool for modeling asynchronous concurrent discrete event processes. A Petri net $N=(\Pi,T,\Phi)$ is a triple, where $\Pi=\{p_1, \ldots ,p_n\}$ is the set of n *places*, $T=\{t_1, \ldots ,t_m\}$ is the set of m *transitions*, and $\Phi \subseteq (\Pi \times T) \cup (T \times T)$ is a set of *arcs* that connect the places and transitions. The *state* of a Petri net or the *marking* ($\mathbf{m}:\Pi \rightarrow \mathbf{N}$) is given by the distribution of *tokens* in its places. Graphically places are represented by circles and transitions are represented by lines perpendicular to the arcs incident on them. Tokens are represented by dark squares that reside in the places. This paper introduces the notion of a T-complement Petri net of a given Petri net . The marking of the T-complement Petri net identifies a maximal set of transitions that can be fired under a corresponding marking of the original Petri net.

A transition is said to be *enabled* if all its input places are marked. An enabled transition can *fire*, which reduces the token load of its input places by one, and increases the token load of its output places by one. For a given marking **m** not all enabled transitions can fire simultaneously. It is possible to identify (possibly non-unique) maximal subsets of transitions that can be fired simultaneously. We use the symbol $\Delta_{max}(\mathbf{m})$ to denote the class of these maximal subsets of transitions, and the symbol $\delta_{max}(\mathbf{m})$ to denote one such maximal set. This concept is illustrated in the following paragraph.

By initializing the Petri net $N=(\Pi,T,\Phi)$ with a marking $\mathbf{m}^0$, it is possible to generate several discrete state trajectories that originate at $\mathbf{m}^0$ by firing enabled transitions. The set of markings that can be found on all such trajectories of length k from $\mathbf{m}^0$ is denoted by $\mathfrak{R}_k(\mathbf{m}^0)$. For a specific trajectory Γ, let $\mathbf{m} \in \mathfrak{R}_k(\mathbf{m}^0)$, $\mathbf{m}_1 \in \mathfrak{R}_1(\mathbf{m})$, and $\mathbf{m}_2 \in \mathfrak{R}_1(\mathbf{m}_1)$ be markings that lie on Γ. The discrete state transition from $\mathbf{m}$ to $\mathbf{m}_1$ is the result of the simultaneous firing of a set of transitions $\delta(\mathbf{m})$, and the transition from $\mathbf{m}_1$ to $\mathbf{m}_2$ is the result of the simultaneous firing of a set of transitions $\delta(\mathbf{m}_1)$. We have $\delta(\mathbf{m}) \subseteq \delta_{max}(\mathbf{m})$, where $\delta_{max}(\mathbf{m}) \in \Delta_{max}(\mathbf{m})$, and $\delta(\mathbf{m}_1) \subseteq \delta_{max}(\mathbf{m}_1)$, where $\delta_{max}(\mathbf{m}_1) \in \Delta_{max}(\mathbf{m}_1)$.

Given a Petri net $N=(\Pi,T,\Phi)$, let $N'=(\Pi',T',\Phi')$ be another Petri net which is initialized with a marking $\mathbf{m}'^0$. Let $P' \subseteq \Pi'$, and $\Theta:T \rightarrow P'$ be a bijection. For the specific discrete state trajectory Γ of N that contains $\mathbf{m},\mathbf{m}_1$, and $\mathbf{m}_2$, suppose that there exists a corresponding trajectory Γ' of N′ that contains the marking $\mathbf{m}' \in \mathfrak{R}_l(\mathbf{m}'^0)$ $(l \geq k)$, and $\mathbf{m}_j' \in \mathfrak{R}_j(\mathbf{m}')$ such that the following properties are satisfied:

$$\mathbf{m}'(p')=\begin{cases} 1 & \text{if } p' \in \Theta(\delta_{max}(\mathbf{m})) \\ 0 & \text{if } p' \in (P'-\Theta(\delta_{max}(\mathbf{m}))) \end{cases} \qquad (1)$$

and,

$$\mathbf{m}'_j(p')=\begin{cases} 1 & \text{if } p' \in \Theta(\delta_{max}(\mathbf{m}_1)) \\ 0 & \text{if } p' \in (P'-\Theta(\delta_{max}(\mathbf{m}_1))) \end{cases} \qquad (2)$$

Furthermore, if $\mathbf{m}'_1, \ldots ,\mathbf{m}'_{j-1}$ is the sequence of markings leading from $\mathbf{m}'$ to $\mathbf{m}'_j$ then $\nexists$

$p' \in (P' - \Theta(\delta_{max}(m_1)))$ such that $m'_k(p') \neq 0$ for all $1 \leq k \leq (j-1)$. Let us also require the above relationship to hold for any k, i.e. for any three successive markings $m \in \Re_k(m^0), m_1 \in \Re_1(m), m_2 \in \Re_1(m_1)$ on Γ.

The net $N' = (\Pi', T', \Phi')$ is said to be a T-complement of $N = (\Pi, T, \Phi)$ if for an initial marking m^0 of N there exists an initial marking m'^0 of N′ such that the relationships (1) and (2) hold for all state trajectories of N originating from m^0 (i.e. for every Γ there exists a Γ' such that the above relationship holds). Note that the definition of a T-complement implicitly requires the specification of $P' \subseteq \Pi'$, and the bijection $\Theta: T \rightarrow P'$.

The Petri net shown in figure 2 is the T-complement of the Petri net shown in figure 1. The places of the T-complement in figure 2 that have the same labels as transitions in the Petri net of figure 1, constitute the set P′. The bijection Θ identifies a place in the Petri net of figure 2 with the same label for every transition in the Petri net of figure 1. As an example of a discrete state trajectory of the Petri net of figure 1 consider a Γ that contains the markings m, m_1, and m_2, where, $m(I_1) = m(E) = m(I_2) = 1$, $m_1(W_1) = m_1(E) = m_1(I_2) = 1$, and $m_2(I_1) = m_2(F) = m_2(I_2) = 1$. The token load of the places not included in the definition of any of the markings is zero. For this Γ there exists a trajectory Γ′ of the Petri net of figure 2 which contains the markings m′, and m'_j, where $m'(t_1) = m'(dp1) = m'(t_{10}) = 1$, and $m'_j(t_2) = m'_j(t_{10}) = 1$. The token load of the places not included in the definition of any of the markings is zero. The relationships (1) and (2) hold for these markings.

For some preliminary results in the theory of Petri nets the reader is referred to the following references [1], [2], [3]. Some results on the construction of T-complement Petri nets for a specific class of Petri nets can be found in the following reference [4].

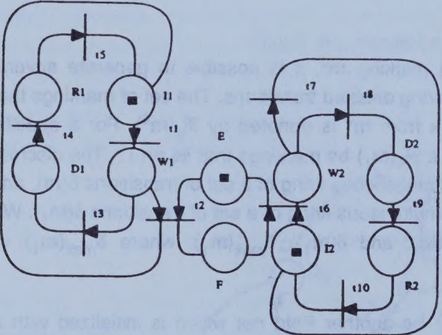


Figure 1: A Petri net

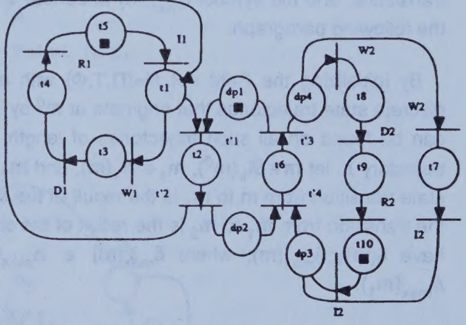


Figure 2: The T-complement Petri net

References

1. Sreenivas, R.S., "Petri Net Synthesis for Conflict Free Resource Allocation in Discrete State Systems", Tech. report, Carnegie-Mellon University, April 1987.

2. Beck, C. L., "Modelling and Simulation of Flexible Control Structures for Automated Manufacturing Systems", Tech. report, Carnegie-Mellon University, December 1985.

3. Peterson, J. L., *Petri net Theory and the Modeling of Systems,* Prentice-Hall, Inc, Englewood Cliffs, NJ 07632, 1981.

4. Sreenivas, R.S., "T-complement Petri Nets", Tech. report, Carnegie-Mellon University, September 1987.

CONTROLLED PETRI NETS AND MAXIMALLY PERMISSIVE FEEDBACK LOGIC

Bruce H. Krogh
Department of Electrical Engineering and Computer Engineering
Carnegie Mellon University
Schenley Park
Pittsburgh, PA 15213

ABSTRACT

Controlled Petri nets (CPNs) are defined as an extension of standard Petri nets with binary control inputs as exogenous conditions for enabling transitions. Properties of the set of reachable markings (states) are established for constant control inputs and the notion of control invariant set of markings is defined for nondynamic state feedback. An algorithm developed by Ramadge and Wonham in the context of controlled automata is described in the present context to compute the maximal control invariant set contained in a given set of admissible markings. We then consider the problem of synthesizing state feedback logic to guarantee the controlled process will remain in a given control invariant set. An algorithm is presented for computing the complete set state feedback laws that are maximally permissive in the sense that they each admit a maximal set of transitions between markings in the control invariant set. The concepts are illustrated by an example.

1. INTRODUCTION

This paper introduces *controlled Petri nets* (CPNs) to model discrete event systems in which the enabling of state transitions can be influenced by exogenous control inputs. CPNs are an extension of standard Petri nets which have been used extensively to model and analyze discrete event systems [1]. When applied to controlled processes, Petri net models necessarily include both the control logic and the internal state transition logic of the process being controlled. These models are useful for simulation and performance analysis for a given control policy. However, to develop a theory of control synthesis for discrete event systems, it is necessary to model the processes independent of the control logic. Toward this end, we define CPNs to model controlled discrete event processes and consider the problem of synthesizing state feedback logic to guarantee the process remains in a given set of admissible states (markings).

The basic framework and results presented in this paper are motivated by, and in part isomorphic to, the work of Ramadge and Wonham on modular feedback logic for controlled automata [2]. CPNs are more general than the automata considered by Ramadge and Wonham in that enabled transitions in the CPN can fire simultaneously, thereby admitting concurrent events, control inputs are not necessarily independent, and there is not a one-to-one correspondence between control inputs and state transitions. As an extension of standard Petri nets, CPNs also provide a compact representation of state transition logic and possess a rich structure for developing a theory of controlled discrete event processes.

In the following section we define CPNs and the firing rule for CPN transitions, and establish some elementary properties of CPN state transitions for constant control inputs. Nondynamic state feedback is defined in section 3 as a mapping of the CPN marking (state) into the set of admissible controls. A set of markings Q is defined to be control invariant if there exists a state feedback such that the state of the CPN remains in Q when the initial marking is in Q. We prove in section 3 the basic result that a set Q is control invariant if and only if it is an invariant of the *zero-input response* of the system. We also describe an algorithm from Ramadge and Wonham for finding the unique maximal control invariant set $\mathcal{A}\uparrow$ contained in a given set $\mathcal{A}$ of admissible markings.

Section 4 concerns the problem of synthesizing a state feedback such that the state of the CPN remains within a given control invariant set Q. We present an algorithm for synthesizing the set of all state feedbacks which are maximally permissive in the sense that they each allow a maximal number of transitions between states in Q. In contrast to the controlled automata considered by Ramadge and Wonham, a maximally permissive state feedback for a CPN is not necessarily unique due to the possibility of concurrent transitions in the CPN. As an example in section 5, we illustrate maximally permissive state feedback for a modified version of the cat-mouse maze from Ramadge and Wonham [2]. Directions for future research are discussed in the concluding section.

2. CONTROLLED PETRI NETS

A *controlled Petri net* (CPN) G is a five-tuple $G=\{P,T,E,C,B\}$ where P is a finite set of *places*, T is a finite set of *transitions*, and $E \subseteq (P \times T) \cup (T \times P)$ is a set of *arcs*, as normally defined for Petri nets [1]. In addition to these standard Petri net elements, we define C as a finite set of *control places* and $B \subseteq C \times T$ as a set of *control arcs* associating control places with transitions. We shall assume that

A1: Each control place is associated with at least one transition, that is, $\forall\ c \in C,\ \exists\ t \in T$ such that $(c,t) \in B$;

A2: No transition is associated with more than one control, that is, if $c,c' \in C$, $c \neq cc'$, and $(c,t) \in B$, then $(c',t) \notin B$.

Thus, the set of transitions T can be partitioned into nonempty subsets of *controlled transitions* $T_c=\{t \in T | (c,t) \in B\}$ for each $c \in C$, and the subset of *uncontrolled transitions*, $T_u=\{t \in T | \forall c \in C,\ (c,t) \notin B\}$.

For $x \in P \cup T$, ${}^\bullet x$ and $x^\bullet$ denote the sets ${}^\bullet x=\{y | (y,x) \in E\}$ and $x^\bullet=\{y | (x,y) \in E\}$, respectively. For a transition $t \in T$, ${}^\bullet t$ is the set of *input places* for t and $t^\bullet$ is the set of *output places* for t. Similarly, for place $p \in P$, ${}^\bullet p$ ($p^\bullet$) is the set of *input* (*output*) *transitions* for p. This notation is extended in the obvious way to denote the set of all input or output transitions (places) for a set of places (transitions).

The *state* of a CPN is given by a *marking* $\mathbf{m}:P \rightarrow N$, where N is the set of nonnegative integers. For each place $p \in P$, $\mathbf{m}(p)$ is the *token load* (the number of *tokens*) in p. We denote the set of all possible markings (i.e., the state space) for a CPN by $\mathcal{M}$ and assume in this paper that

A3: The set $\mathcal{M}$ of possible markings is finite.

We note that in many applications it as possible to establish *a priori* that the state space for a CPN is finite using, for example, the theory of invariants for Petri nets. Moreover, removing assumption A3 introduces only minor changes in the results of this paper.

A *control* $\mathbf{u}:C \rightarrow \{0,1\}$ assigns a token load of 0 or 1 to each control place. Given controls $\mathbf{u}^1,\mathbf{u}^2$, we use the following notation and definitions to partially order and construct controls:

$\mathbf{u}^1 \leq \mathbf{u}^2$ iff $\forall\ c \in C\ \mathbf{u}^1(c) \leq \mathbf{u}^2(c)$,

$\mathbf{u}^1 < \mathbf{u}^2$ iff $\mathbf{u}^1 \leq \mathbf{u}^2$ and $\exists\ c \in C$ such that $\mathbf{u}^1(c) < \mathbf{u}^2(c)$,

$\mathbf{u}^1 \wedge \mathbf{u}^2$ (*conjunction*) defines the control $\mathbf{u}=\mathbf{u}^1 \wedge \mathbf{u}^2$ where $\mathbf{u}(c)=1$ iff $\mathbf{u}^1(c)=\mathbf{u}^2(c)=1$,

$\mathbf{u}^1 \vee \mathbf{u}^2$ (*disjunction*) defines the control $\mathbf{u}=\mathbf{u}^1 \vee \mathbf{u}^2$ where $\mathbf{u}(c)=0$ iff $\mathbf{u}^1(c)=\mathbf{u}^2(c)=0$.

We denote the set of admissible controls by $\mathcal{U}$ where $\mathcal{U}$ satisfies the following assumptions:

A4: $\mathbf{0} \in \mathcal{U}$, where $\mathbf{0}$ is the control which is zero for all $c \in C$;

A5: $\mathcal{U}$ is closed under conjunction and disjunction, that is, if $\mathbf{u}^1,\mathbf{u}^2 \in \mathcal{U}$ then $\mathbf{u}^1 \wedge \mathbf{u}^2 \in \mathcal{U}$ and $\mathbf{u}^1 \vee \mathbf{u}^2 \in \mathcal{U}$;

A6: For every $c \in C$ there exists a $\mathbf{u} \in \mathcal{U}$ such that $\mathbf{u}(c)=1$.

Assumption A5 is equivalent to assuming $\mathcal{U}$ is a *lattice* with respect to the partial ordering " $\leq$ " of controls defined above. Assumptions A5 and A6 together imply that $\mathbf{1} \in \mathcal{U}$, where $\mathbf{1}$ is the control that is 1 for all $c \in C$.

As with standard Petri nets, state transitions in a CPN occur when *enabled* transitions *fire*, where these terms are defined for CPNs as follows. Given a marking $\mathbf{m}$, a transition $t \in T$ is said to be *state enabled* when the token load of each of its input places is at least one, that is, when $\mathbf{m}(p) \geq 1$ $\forall\, p \in {}^\bullet t$. Given a control $\mathbf{u}$, a controlled transition $t \in T_c$ associated with control place $c \in C$ is said to be *control enabled* when $\mathbf{u}(c)=1$. A transition is *enabled* when it is both state enabled and control enabled, where uncontrolled transitions (transitions in T_u) are control enabled by definition. For a given marking $\mathbf{m}$ and control $\mathbf{u}$, the set of enabled transitions will be denoted by $T_e(\mathbf{m},\mathbf{u})$.

Enabled transitions in $T_e(\mathbf{m},\mathbf{u})$ can fire, in which case the state changes from marking $\mathbf{m}$ to a new marking $\mathbf{m}'$ defined by the *CPN firing rule* given below. The time at which enabled transitions fire is not specified in the present CPN definition, but is assumed that when transitions fire the resulting state transitions occur instantaneously.[1] When more than one transition is enabled for a given marking $\mathbf{m}$ and control $\mathbf{u}$, more than one transition in $T_e(\mathbf{m},\mathbf{u})$ may fire at a given instant, provided the token load is sufficient for input places shared by the transitions. This is made precise in the following firing rule for CPNs:

CPN firing rule: Given a CPN G with a marking $\mathbf{m}$ and control $\mathbf{u}$, a set $T_s \subseteq T_e(\mathbf{m},\mathbf{u})$ of enabled transitions can fire simultaneously provided

$$|\mathbf{m}(p)| \geq |p^\bullet \cap T_s| \qquad \forall\ p \in {}^\bullet T_s, \tag{1}$$

where $|\cdot|$ denotes the cardinality of the set argument. If inequality (1) is satisfied and the transitions in T_s fire simultaneously, the CPN marking is updated from $\mathbf{m}$ to a marking $\mathbf{m}'$ defined for each $p \in P$ by

$$\mathbf{m}'(p) = \mathbf{m}(p) - |p^\bullet \cap T_s| + |{}^\bullet p \cap T_s|. \tag{2}$$

We refer to transitions which fire simultaneously as *concurrent transitions*. Concurrent transitions in a CPN result in a single *state transition* defined by equation (2). In the sequel the notation $(\mathbf{m},\mathbf{u}) \rightarrow T_s \rightarrow \mathbf{m}'$ indicates that state $\mathbf{m}'$ results from simultaneously firing the transitions in T_s, where it is assumed that T_s is a subset of $T_e(\mathbf{m},\mathbf{u})$ and inequality (1) is satisfied. Note that inequality (1) guarantees that the marking defined by (2) is valid, that is, the token load for every place is nonnegative.

We make the following assumption concerning the uniqueness of concurrent transitions which can lead to a particular state transition:

A7: Given a marking and control $(\mathbf{m},\mathbf{u})$, sets of enabled transitions T_1, T_2, and a marking $\mathbf{m}'$, if $(\mathbf{m},\mathbf{u}) \rightarrow T_1 \rightarrow \mathbf{m}'$ and $(\mathbf{m},\mathbf{u}) \rightarrow T_2 \rightarrow \mathbf{m}'$, then $T_1 = T_2$.

A CPN which satisfies assumption A7 shall be called a *minimal realization* of a discrete event process.

For a given marking and control $(\mathbf{m},\mathbf{u})$, the CPN firing rule determines the *state transition set* $\delta(\mathbf{m},\mathbf{u})$ which is the set of all markings which could result from concurrent transitions in $T_e(\mathbf{m},\mathbf{u})$. We note that if $T_e(\mathbf{m},\mathbf{u}) = \emptyset$, the empty set, then $\delta(\mathbf{m},\mathbf{u}) = \emptyset$. The following properties of $\delta(\mathbf{m},\mathbf{u})$ will be used in the sequel.

Lemma 1. Given a marking $\mathbf{m}$ and control $\mathbf{u}$, if $T_e(\mathbf{m},\mathbf{u}) \neq \emptyset$, then $\delta(\mathbf{m},\mathbf{u}) \neq \emptyset$.

Proof. The lemma follows from the observation that any singleton subset of $T_e(\mathbf{m},\mathbf{u})$ satisfies (1), implying $\delta(\mathbf{m},\mathbf{u}) \neq \emptyset$.

Lemma 2. Given a marking $\mathbf{m}$ and controls $\mathbf{u}^1, \mathbf{u}^2$, if $\mathbf{u}^1 \leq \mathbf{u}^2$, then $\delta(\mathbf{m},\mathbf{u}^1) \subseteq \delta(\mathbf{m},\mathbf{u}^2)$.

---

[1] In an extension of the present CPN definition to *timed* CPNs, stochastic or deterministic delays would be assigned to places rather than transitions.

Proof. If $T_e(\mathbf{m},\mathbf{u}^1)=\emptyset$, then $\delta(\mathbf{m},\mathbf{u})=\emptyset$ and the lemma is trivially true. If $T_e(\mathbf{m},\mathbf{u}^1)\neq\emptyset$, then $\delta(\mathbf{m},\mathbf{u})\neq\emptyset$ by lemma 1. For any $\mathbf{m}'\in\delta(\mathbf{m},\mathbf{u}^1)$ there exists a set of concurrent transitions $T_s\subseteq T_e(\mathbf{m},\mathbf{u}^1)$ such that $(\mathbf{m},\mathbf{u}^1)\rightarrow T_s\rightarrow\mathbf{m}'$. Now, $\mathbf{u}^1\leq\mathbf{u}^2$ implies that $T_e(\mathbf{m},\mathbf{u}^1)\subseteq T_e(\mathbf{m},\mathbf{u}^2)$ since any transition which is control enabled by $\mathbf{u}^1$ is necessarily control enabled by $\mathbf{u}^2$. This implies that $T_s\subseteq T_e(\mathbf{m},\mathbf{u}^2)$. Moreover, T_s is a set of transitions which can fire simultaneously under $(\mathbf{m},\mathbf{u}^2)$ since inequality (1) is satisfied for $\mathbf{m}$ and T_s. Therefore, $(\mathbf{m},\mathbf{u}^2)\rightarrow T_s\rightarrow\mathbf{m}'$, implying $\mathbf{m}'\in\delta(\mathbf{m},\mathbf{u}^2)$, which proves the lemma.

Lemma 3. For any marking $\mathbf{m}$ and controls $\mathbf{u}^1,\mathbf{u}^2$, $\delta(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)=\delta(\mathbf{m},\mathbf{u}^1)\cap\delta(\mathbf{m},\mathbf{u}^2)$.

Proof. Since $\mathbf{u}^1\wedge\mathbf{u}^2\leq\mathbf{u}^j$ for j=1,2, it follows from lemma 2 that $\delta(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)\subseteq\delta(\mathbf{m},\mathbf{u}^1)\cap\delta(\mathbf{m},\mathbf{u}^2)$. Now, suppose $\mathbf{m}'\in\delta(\mathbf{m},\mathbf{u}^1)\cap\delta(\mathbf{m},\mathbf{u}^2)$. Then there exist sets of concurrent transitions $T_1\subseteq T_e(\mathbf{m},\mathbf{u}^1)$ and $T_2\subseteq T_e(\mathbf{m},\mathbf{u}^2)$ such that $(\mathbf{m},\mathbf{u}^1)\rightarrow T_1\rightarrow\mathbf{m}'$ and $(\mathbf{m},\mathbf{u}^2)\rightarrow T_2\rightarrow\mathbf{m}'$. Consider the control $\mathbf{u}^1\vee\mathbf{u}^2$. Clearly, $T_j\subseteq T_e(\mathbf{m},\mathbf{u}^1\vee\mathbf{u}^2)$ for j=1,2. Moreover, the transitions in T_j can fire simultaneously since (1) is satisfied for $\mathbf{m}$. Therefore

$$(\mathbf{m},\mathbf{u}^1\vee\mathbf{u}^2)\rightarrow T_j\rightarrow\mathbf{m}' \qquad \text{for } j=1,2. \tag{3}$$

Assuming the CPN is a minimal realization (A7), equation (3) implies $T_1=T_2\equiv T_s$. But this implies $T_s\subseteq T_e(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)$ since all transitions in T_s are control enabled for both $\mathbf{u}^1$ and $\mathbf{u}^2$. Therefore, since (1) is satisfied for T_s and $\mathbf{m}$, we have $(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)\rightarrow T_s\rightarrow\mathbf{m}'$. This implies $\mathbf{m}'\in\delta(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)$, which proves $\delta(\mathbf{m},\mathbf{u}^1)\cap\delta(\mathbf{m},\mathbf{u}^2)\subseteq\delta(\mathbf{m},\mathbf{u}^1\wedge\mathbf{u}^2)$, thereby proving the lemma.

Lemma 4. Given controls $\mathbf{u}^1,\mathbf{u}^2$ and a marking $\mathbf{m}$, $\delta(\mathbf{m},\mathbf{u}^1\vee\mathbf{u}^2)\supseteq\delta(\mathbf{m},\mathbf{u}^1)\cup\delta(\mathbf{m},\mathbf{u}^2)$.

Proof. The lemma follows immediately from lemma 2 and the fact that for j=1,2, $\mathbf{u}^j\leq\mathbf{u}^1\vee\mathbf{u}^2$.

Remark. We note that the state transition sets in lemma 4 are not necessarily equal because concurrent transitions can exist for control $\mathbf{u}^1\vee\mathbf{u}^2$ which cannot fire simultaneously under $\mathbf{u}^1$ or $\mathbf{u}^2$ alone. Therefore, there can be markings which are reachable from $\mathbf{m}$ in a single transition under $b[u]^1\vee\mathbf{u}^2$ which are not reachable in a single state transition under the individual controls.

Repeated firing of enabled transitions results in a *state trajectory* for the CPN where each marking along the state trajectory is in the state transition set of the previous marking and control. Given a set of markings $\mathcal{M}_o$ and a control $\mathbf{u}$, $R_{\mathbf{u}}(\mathcal{M}_o)$ denotes the set of *reachable markings* for constant control $\mathbf{u}$ and initial markings in $\mathcal{M}_o$, that is, $R_{\mathbf{u}}(\mathcal{M}_o)$ is the set of all markings which occur along all possible state trajectories of the CPN starting from, and including, an initial marking $\mathbf{m}_o\in\mathcal{M}_o$ with control $\mathbf{u}$. As a special case, when $\mathbf{u}=\mathbf{0}$, we refer to $R_{\mathbf{0}}(\mathcal{M}_o)$ as the *zero-input response* of the CPN with respect to the initial markings in $\mathcal{M}_o$. Finally, we note that it follows by induction from lemma 2 that if two controls $\mathbf{u},\mathbf{u}'$ are related as $\mathbf{u}\leq\mathbf{u}'$, then $\mathbf{R}_{\mathbf{u}}(\mathcal{M}_o)\subseteq\mathbf{R}_{\mathbf{u}'}(\mathcal{M}_o)$.

3. STATE FEEDBACK AND CONTROL INVARIANTS

A *state feedback* for a CPN G is a function $\mathbf{F}:\mathcal{M}\rightarrow\mathcal{U}$ which assigns an admissible control to every marking of G. Given a state feedback $\mathbf{F}$ and a control place $c\in C$, $\mathbf{F}_c:\mathcal{M}\rightarrow\{0,1\}$ denotes the control defined by $\mathbf{F}$ for the control place c, that is, given $\mathbf{m}\in\mathcal{M}$, if $\mathbf{u}=\mathbf{F}(\mathbf{m})$, then $\mathbf{F}_c(\mathbf{m})=\mathbf{u}(c)$. For each state feedback $\mathbf{F}$ we assume:

A8: If a state transition occurs from marking $\mathbf{m}$ to marking $\mathbf{m}'$, the control is updated instantaneously from $\mathbf{F}(\mathbf{m})$ to $\mathbf{F}(\mathbf{m}')$; and

A9: If no transitions in T_c are state enabled for some $c\in C$, that is, if $T_c\cap T_e(\mathbf{m},\mathbf{1})=\emptyset$, then

$\mathbf{F}_c \equiv 0$.

We note that in the absence of explicit timing in the CPN definition, an assumption like A8 is necessary for the state trajectories to be well defined. We also note that assumption A9 introduces no loss of generality when the entire state (marking) can be measured.

When a state feedback $\mathbf{F}$ is applied to CPN G, the resulting closed-loop process will be denoted by $G_{\mathbf{F}}$. By assumption A8, the state transition set for a marking $\mathbf{m}$ in $G_{\mathbf{F}}$ is given by $\delta(\mathbf{m},\mathbf{F}(\mathbf{m}))$. The set of markings which can be reached along state trajectories in $G_{\mathbf{F}}$ starting from an initial marking in a set $\mathcal{M}_o \subset \mathcal{M}$ will be denoted by $R_{\mathbf{F}}(\mathcal{M}_o)$.

We introduce a partial ordering on the set $\mathcal{F}$ of state feedbacks for a CPN G as follows. For $\mathbf{F},\mathbf{F}' \in \mathcal{F}$:

$\mathbf{F} \leq \mathbf{F}'$ iff $\mathbf{F}(\mathbf{m}) \leq \mathbf{F}'(\mathbf{m})$ for every $\mathbf{m} \in \mathcal{M}$;

$\mathbf{F} < \mathbf{F}'$ iff $\mathbf{F} \leq \mathbf{F}'$ and $\mathbf{F}(\mathbf{m}) < \mathbf{F}'(\mathbf{m})$ for some $\mathbf{m} \in \mathcal{M}$.

A state feedback $\mathbf{F}'$ is said to be *more permissive* than a state feedback $\mathbf{F}$ when $\mathbf{F} < \mathbf{F}'$. It follows from lemma 2 that if $\mathbf{F}'$ is more permissive than $\mathbf{F}$, then $\delta(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq \delta(\mathbf{m},\mathbf{F}'(\mathbf{m}))$ for all $\mathbf{m} \in \mathcal{M}$ and $R_{\mathbf{F}}(\mathcal{M}_o) \subseteq R_{\mathbf{F}'}(\mathcal{M}_o)$ for any $\mathcal{M}_o \subset \mathcal{M}$.

A set of markings $Q \subset \mathcal{M}$ is said to be *invariant* with respect to a state feedback $\mathbf{F}$ (or, $\mathbf{F}$-*invariant*) if for every $\mathbf{m} \in Q$, $\delta(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq Q$. In words, a set Q is $\mathbf{F}$-invariant when only markings in Q can be reached in one transition from markings in Q. The following lemma gives an equivalent definition for $\mathbf{F}$-invariants in terms of the set of reachable markings.

Lemma 5. A set $Q \subset \mathcal{M}$ is $\mathbf{F}$-invariant for a state feedback $\mathbf{F}$ iff $Q = R_{\mathbf{F}}(Q)$.

Proof. $Q \subseteq R_{\mathbf{F}}(Q)$ by the definition of $R_{\mathbf{F}}(Q)$. For a marking $\mathbf{m}$ and $k \geq 0$, let $\delta_k(\mathbf{m},\mathbf{F}(\mathbf{m}))$ denote the set of markings reachable in exactly k state transitions starting from $\mathbf{m}$, with $\delta_0(\mathbf{m},\mathbf{F}(\mathbf{m})) = \{\mathbf{m}\}$. Clearly $R_{\mathbf{F}}(Q)$ is equal to the union of the sets $\delta_k(\mathbf{m},\mathbf{F}(\mathbf{m}))$ over all $\mathbf{m}$ in Q and $k=0,1,2,\ldots$. We now show by induction that for every $\mathbf{m} \in Q$ and every $k \geq 0$, $\delta_k(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq Q$. Clearly this is true for every $\mathbf{m} \in Q$ when $k=0$. Suppose for a given $k \geq 0$ and $\mathbf{m} \in Q$, $\delta_k(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq Q$. Then, for any $\mathbf{m}' \in \delta_k(\mathbf{m},\mathbf{F}(\mathbf{m}))$, $\delta_1(\mathbf{m}',\mathbf{F}(\mathbf{m}')) \subseteq Q$. But $\delta_{k+1}(\mathbf{m},\mathbf{F}(\mathbf{m}))$ is equal to the union of the sets $\delta_1(\mathbf{m}',\mathbf{F}(\mathbf{m}'))$ over the states $\mathbf{m}' \in \delta_k(\mathbf{m},\mathbf{F}(\mathbf{m}))$. Therefore, $\delta_{k+1}(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq Q$, which proves the lemma.

A set of markings Q is said to be *control invariant* if it is $\mathbf{F}$-invariant for some state feedback $\mathbf{F}$. The following proposition states that a set of markings is control invariant if and only if it is invariant with respect to the null feedback, $\mathbf{F} \equiv \mathbf{0}$.

Proposition 1. A set of markings Q is control invariant iff Q is $\mathbf{0}$-invariant.

Proof. If: By definition.

Only if: Suppose Q is $\mathbf{F}$-invariant for some state feedback $\mathbf{F} \not\equiv \mathbf{0}$. Then by lemma 5, $Q = R_{\mathbf{F}}(Q)$. Since $\mathbf{F}$ is more permissive than $\mathbf{0}$, $R_{\mathbf{0}}(Q) \subseteq R_{\mathbf{F}}(Q) = Q$. But $Q \subseteq R_{\mathbf{0}}(Q)$ by definition. Therefore, $Q = R_{\mathbf{0}}(Q)$ which by lemma 5 proves the proposition.

Proposition 1, which is the CPN analogue of Proposition 6.1 in reference [2], implies that only the zero-input response of the system needs to be considered to test whether a set of markings is control invariant. However, once it is determined that a set is control invariant, it is desirable to find a state feedback that is more permissive than $\mathbf{F}=\mathbf{0}$, since this is the most restrictive control policy. In the following section we consider the problem of constructing maximally permissive state feedback for a given control invariant set. We note from lemma 5 that more permissive state feedback does not increase the set of markings reachable from a control invariant Q; more permissive state feedback merely increases the number of possible transitions between markings in Q.

To conclude this section we consider the following problem. Suppose a state feedback is to be designed to guarantee that the CPN marking remains in a given set of *admissible markings*, $\mathcal{A}$, provided the initial marking is in $\mathcal{A}$. If $\mathcal{A}$ is not control invariant, no such state feedback exists. In this case it is desirable to find, if possible, the *maximal control invariant set* $\mathcal{A}\uparrow$ in $\mathcal{A}$, that is, the set $\mathcal{A}\uparrow \subseteq \mathcal{A}$ which is control invariant and contains all other control invariant sets in $\mathcal{A}$.

The existence and uniqueness of the maximal control invariant set for a given set of admissible states was proved by Ramadge and Wonham in the context of controlled automata [2]. In the present context it is easily shown that $\mathcal{A}\uparrow$ exists (it may be the null set) and is unique, by observing that the class of control invariant sets is closed under unions. Ramadge and Wonham present an algorithm to compute maximal control invariant sets for controlled automata [2]. For completeness we restate their algorithm using the terminology of CPNs and the following definition.

Given a CPN G, a set of markings Q and a state feedback $\mathbf{F}$, the *antecedent of* Q under state feedback $\mathbf{F}$ is defined as the set of markings $\Phi_{\mathbf{F}}(Q)$ given by

$$\Phi_{\mathbf{F}}(Q)=\{\mathbf{m}\in \mathcal{M} \mid \delta(\mathbf{m},\mathbf{F}(\mathbf{m}))\subseteq Q\}.$$

In words, the antecedent of Q is the set of markings from which only markings in Q can be reached in a single state transition under the state feedback $\mathbf{F}$. The antecedent of Q is the set-theoretic analogy to the logical notion of the *weakest liberal predicate* [2].

Given a set $\mathcal{A}$ of admissible markings, Ramadge and Wonham show that the maximal control invariant set $\mathcal{A}\uparrow$ contained in $\mathcal{A}$ can be computed by performing the following iteration

$$\mathcal{A}_{k+1} = \mathcal{A}_k \cap \Phi_{\mathbf{0}}(\mathcal{A}_k), \tag{4}$$

with $\mathcal{A}_0=\mathcal{A}$. The iteration terminates when $\mathcal{A}_{k+1}=\mathcal{A}_k\equiv\mathcal{A}\uparrow$. In the present context, assumption A3 guarantees that the iteration will converge. Further details concerning iteration (4) can be found in reference [2].

4. MAXIMALLY PERMISSIVE FEEDBACK LOGIC

In this section we will be concerned with the definition of state feedbacks for markings in a given control invariant set. Thus, for simplicity of notation and without loss of generality, we assume:

A10: All state feedbacks are defined to be the null control, $\mathbf{0}$, for markings outside the given control invariant set.

Given a set of markings Q which is control invariant we define $\mathcal{F}(Q)$ as the set of state feedbacks given by

$$\mathcal{F}(Q) = \{\mathbf{F}\in \mathcal{F} \mid Q \text{ is } \mathbf{F}\text{-invariant}\}.$$

The set $\mathcal{F}(Q)$ is partially ordered by the order relation defined in the previous section, and it follows from lemma 3 that $\mathcal{F}(Q)$ is closed under conjunction, where for $\mathbf{F}_1,\mathbf{F}_2\in\mathcal{F}$, $\mathbf{F}^1\wedge\mathbf{F}^2$ is defined for each $\mathbf{m}\in\mathcal{M}$ as $\mathbf{F}^1(\mathbf{m})\wedge\mathbf{F}^2(\mathbf{m})$. A maximal element in $\mathcal{F}(Q)$ is said to be *maximally permissive* with respect to Q. In other words, a state feedback $\mathbf{F}$ is maximally permissive with respect to Q if there is no state feedback more permissive than $\mathbf{F}$ for which Q is invariant. We note that there can be more than one maximal element in $\mathcal{F}(Q)$ due to the possibility of concurrent transitions in a CPN. In fact, $\mathcal{F}(Q)$ is not necessarily closed under disjunction (see the remark following lemma 4). Thus, there is not necessarily a *unique* maximally permissive state feedback. In the following we present an algorithm for finding the set of *all* maximally permissive state feedbacks for a given control invariant set Q.

To facilitate the presentation of the algorithm, we index the control places as $C=\{c_1, \ldots, c_r\}$ where $r=|C|$ and for $j=1,\ldots,r$ we define the control $\mathbf{u}_j\in\mathcal{U}$ as the minimal element in the set

$\{\mathbf{u} \in \mathcal{U} \mid \mathbf{u}(c_j)=1\}$. Existence and uniqueness of the controls $\mathbf{u}_j$ follow from assumptions A4-A6.

Given a control invariant set Q, we define set of state feedbacks $\mathcal{F}\uparrow(Q)$ as

$$\mathcal{F}\uparrow(Q) \;=\; \{\mathbf{F} \;\mid\; \mathbf{F}(\mathbf{m}) \in \mathcal{U}\uparrow(Q,\mathbf{m}) \;\;\forall\;\; \mathbf{m} \in Q\}, \tag{5}$$

where for each marking $\mathbf{m} \in Q$ the set of controls $\mathcal{U}\uparrow(Q,\mathbf{m})$ is generated by the following steps:

1. Let $\mathcal{U}_1(Q,\mathbf{m})$ be the set of controls $\mathbf{u}_j$, for $j \in \{1,\ldots,r\}$, such that
    a. $T_{c_j} \cap T_e(\mathbf{m},\mathbf{1}) \neq \emptyset$; and
    b. $\delta(\mathbf{m},\mathbf{u}_j) \subseteq Q$.

    Set $K=|\mathcal{U}_1(Q,\mathbf{m})|$

2. If $K=0$, then set $\mathcal{U}\uparrow(Q,\mathbf{m})=\{\mathbf{0}\}$, and stop

    If $K > 0$, set $\mathcal{U}\uparrow(Q,\mathbf{m})=\emptyset$ and $k=1$, and continue.

3. Let $\mathcal{U}_{k+1}(Q,\mathbf{m})$ be the set of controls $\mathbf{u}$ satisfying
    a. $\mathbf{u} \;=\; \mathbf{u}^k \vee \mathbf{u}^1$ where $\mathbf{u}^k \in \mathcal{U}_k(Q,\mathbf{m})$ and $\mathbf{u}^1 \in \mathcal{U}_k(Q,\mathbf{m})$; and
    b. $\delta(\mathbf{m},\mathbf{u}) \subseteq Q$.

4. Add to $\mathcal{U}\uparrow(Q,\mathbf{m})$ all $\mathbf{u} \in \mathcal{U}_k(Q,\mathbf{m})$ for which $\mathbf{u} \wedge \mathbf{u}' \neq \mathbf{u} \;\forall\; \mathbf{u}' \in \mathcal{U}_{k+1}(Q,\mathbf{m})$.

5. Eliminate from $\mathcal{U}_1(Q,\mathbf{m})$ all controls $\mathbf{u}_j$ for which $\mathbf{u}_j \wedge \mathbf{u}=0 \;\forall\; \mathbf{u} \in \mathcal{U}_{k+1}(Q,\mathbf{m})$.

6. If $k < K$ and $\mathcal{U}_{k+1}(Q,\mathbf{m}) \neq \emptyset$, increment k and go to step 3; otherwise stop.

We claim $\mathcal{F}\uparrow(Q)$ defined by equation (5) and steps 1-6 above is the complete set of maximally permissive state feedbacks. We first prove the following lemma.

Lemma 6. Suppose steps 1-6 are executed for a set of markings Q which is control invariant and $\mathbf{m} \in Q$. If for some $\mathbf{u} \in \mathcal{U}$, $\delta(\mathbf{m},\mathbf{u}) \subseteq Q$ and $\mathbf{u}(c_1)+ \cdots \mathbf{u}(c_r)=k$, then $\mathbf{u} \in \mathcal{U}_k(Q,\mathbf{m})$.

Proof. For a binary row vector $\mathbf{b}=[b_1, \ldots, b_r] \in \{0,1\}^r$, we define the control $\mathbf{u}_b$ as

$$\mathbf{u}_b \;=\; b_1\mathbf{u}_1 \vee \cdots \vee b_r\mathbf{u}_r. \tag{6}$$

Let the components of $\mathbf{b} \in \{0,1\}^r$ be given by $b_j=\mathbf{u}(c_j)$ for $j=1,\ldots,r$. We first show that $\mathbf{u}=\mathbf{u}_b$. By equation (6), $\mathbf{u} \leq \mathbf{u}_b$ since $\mathbf{u}_j(c_j)=1$ for $j=1,\ldots,r$. Suppose $\mathbf{u} < \mathbf{u}_b$, that is, $\exists\ j' \in \{1,\ldots r\}$ such that $\mathbf{u}(c_{j'})=0$ and $\mathbf{u}_b(c_{j'})=1$. Then equation (6) and the definition of $\mathbf{b}$ imply $\exists\ j'' \in \{1,\ldots r\}$, $j'' \neq j'$, such that $\mathbf{u}(c_{j''})=1$ and $\mathbf{u}_{j''}(c_{j'})= 1$. But by the definition of $\mathbf{u}_{j''}$ we must have $\mathbf{u}_{j''}=\mathbf{u}_{j''} \wedge \mathbf{u}$, which contradicts the assumption that $\mathbf{u}(c_{j'})=0$, thereby proving $\mathbf{u}=\mathbf{u}_b$. Now, for any $\mathbf{b}' \in \{0,1\}^r$ such that $\mathbf{b}' < \mathbf{b}$, lemma 2 implies $\delta(\mathbf{m},\mathbf{u}_{b'}) \subseteq \delta(\mathbf{m},\mathbf{u}) \subseteq Q$. Thus, by induction on the number n of non-zero elements in $\mathbf{b}' \leq \mathbf{b}$ it can be shown that $\mathbf{u}_{b'} \in \mathcal{U}_n(Q,\mathbf{m})$ for $n=1,\ldots,k$, thereby proving the lemma.

Proposition 2. Given a CPN and a control invariant set of markings Q, a state feedback $\mathbf{F}$ is maximally permissive iff $\mathbf{F} \in \mathcal{F}\uparrow(Q)$, where $\mathcal{F}\uparrow(Q)$ is given by equation (5).

Proof. If: Suppose $\mathbf{F} \in \mathcal{F}\uparrow(Q)$. Then from equation (5) for any $\mathbf{m} \in Q$, $\mathbf{F}(\mathbf{m}) \in \mathcal{U}\uparrow(Q,\mathbf{m})$. Thus, by conditions 2.b and 3.b in the steps above, $\delta(\mathbf{m},\mathbf{F}(\mathbf{m})) \subseteq Q$, which implies Q is $\mathbf{F}$-invariant. Suppose $\mathbf{F}$ is not maximally permissive, that is, suppose $\exists\ \mathbf{F}' \in \mathcal{F}(Q)$ such that $\mathbf{F} < \mathbf{F}'$. Then, for some $\mathbf{m} \in Q$, $\mathbf{F}(\mathbf{m}) < \mathbf{F}'(\mathbf{m})$, which implies that if $k=\mathbf{F}_{c_1}(\mathbf{m})+ \cdots +\mathbf{F}_{c_r}(\mathbf{m})$ and

$k' = \mathbf{F}_{c_1}'(\mathbf{m}) + \cdots + \mathbf{F}_{c_r}'(\mathbf{m})$, then $k < k'$. By lemma 6, $\mathbf{F}'(\mathbf{m}) \in \mathcal{U}_{k'}(\mathcal{Q},\mathbf{m})$, which implies $\exists$ $\mathbf{u} \in \mathcal{U}\!\uparrow_{k+1}(\mathcal{Q},\mathbf{m})$ such that $\mathbf{F}(\mathbf{m}) < \mathbf{u} \leq \mathbf{F}'(\mathbf{m})$. But this contradicts the assumption that $\mathbf{F} \in \mathcal{F}\!\uparrow(\mathcal{Q})$ since such a control would not have satisfied the condition in step 4 for k defined above. Therefore **F** is a maximal element of $\mathcal{F}(\mathcal{Q})$.

ONLY IF: Suppose **F** is maximally permissive but not in $\mathcal{F}\!\uparrow(\mathcal{Q})$. There cannot be any element of $\mathcal{F}\!\uparrow(\mathcal{Q})$ which is more permissive than **F**, since it is shown above that $\mathcal{F}\!\uparrow(\mathcal{Q}) \subseteq \mathcal{F}(\mathcal{Q})$. Moreover, **F** cannot be more permissive than any element of $\mathcal{F}\!\uparrow(\mathcal{Q})$ since we have shown all elements of $\mathcal{F}\!\uparrow(\mathcal{Q})$ are maximally permissive. Therefore, **F** cannot be related to *any* elements of $\mathcal{F}\!\uparrow(\mathcal{Q})$. However, lemma 6 implies that for every $\mathbf{m} \in \mathcal{Q}$, $\exists\ k_{\mathbf{m}} \in \{0, \ldots, r\}$ such that $\mathbf{F}(\mathbf{m}) \in \mathcal{U}_{k_{\mathbf{m}}}(\mathcal{Q},\mathbf{m})$, which in turn implies $\exists\ \mathbf{u}_{\mathbf{m}} \in \mathcal{U}\!\uparrow(\mathcal{Q},\mathbf{m})$ such that $\mathbf{F}(\mathbf{m}) \leq \mathbf{u}_{\mathbf{m}}$. Defining state feedback $\mathbf{F}'$ for each $\mathbf{m} \in \mathcal{Q}$ as $\mathbf{F}'(\mathbf{m}) = \mathbf{u}_{\mathbf{m}}$, we have $\mathbf{F} \leq \mathbf{F}' \in \mathcal{F}\!\uparrow(\mathcal{Q})$, contradicting the assumption that **F** is maximally permissive, and thereby proving the proposition.

5. EXAMPLE

To illustrate the application of CPNs to model controlled discrete event processes and the synthesis of maximally permissive state feedback we consider a modification of the cat-mouse maze in reference [2]. The "maze" shown in figure 1 has five rooms, R1,...,R5, and six controlled doors, D1,...,D6, where each door can be set at one of three positions: fully opened, partially opened, and closed. The opening between rooms R1 and R2 is fully opened at all times. One cat and one mouse are running loose in the maze. The cat can only pass through doors that are fully opened, while the mouse can go through a partially or fully opened door. The objective is to control the doors so that the cat and mouse never occupy the same room, assuming they start in different rooms and that their locations can always be detected.

A CPN model for this example is shown in figure 2 consisting of two sub-CPNs each with one token representing the location of the cat and mouse, respectively. As with standard Petri nets, circles represent places, and bars represent transitions. The places labeled CR1,...,CR5 (MR1,...,MR5) in figure 2 correspond to the cat (mouse) being in room R1,...,R5, respectively. The control places, represented by squares labeled F1,...,F6 (P1,...,P6) correspond to the door states of being fully (partially) opened. We note that, in contrast to the example of Ramadge and Wonham, the control for a particular CPN transition does not correspond to the control of a single state transition. The cat and mouse can go through doors at the same instant, resulting in current transitions in the CPN. Also, the controls for each CPN transition are not independent; opening a door for the cat will also let the mouse pass through. Thus, the set $\mathcal{U}$ of admissible controls is given by

$$\mathcal{U} = \{\mathbf{u} \mid \text{for } j=1,\ldots,6 \text{ if } \mathbf{u}(Fj) = 1, \text{ then } \mathbf{u}(Pj) = 1\}.$$

The set $\mathcal{M}$ of all possible markings for the CPN finite and the set $\mathcal{A}$ of admissible markings is given by

$$\mathcal{A} = \{\mathbf{m} \mid \forall\, j=1,\ldots,5,\ \mathbf{m}(CRj)\mathbf{m}(MRj) = 0\}.$$

Applying the algorithm from section 3 to compute $\mathcal{A}\!\uparrow$ gives (as in [2])

$$\mathcal{A}\!\uparrow = \mathcal{A} - \{\mathbf{m} \mid \mathbf{m}(CR1) = \mathbf{m}(MR2) = 1 \text{ or } \mathbf{m}(CR2) = \mathbf{m}(MR1) = 1\}.$$

Applying the algorithm of the previous section we can find the set of all maximally permissive state feedbacks with respect to $\mathcal{A}\!\uparrow$. For example, suppose the cat is in room R3 and the mouse is in room R5. Then doors D2 and D3 remain closed since the corresponding control transitions are not state enabled. On the other hand, doors D1 and D5 can be fully opened by control places F1 and F5, respectively, and doors D4 and D6 can be partially opened by control places P4 and P6, respectively. However, if F5 and P6 are enabled, then the cat and mouse can enter room R4 simultaneously, which is a forbidden state. If F1 and P4 are enabled simultaneously, then the cat can enter room R1 and the mouse can enter room R2, which is a state which is in $\mathcal{A}$ but not in the control invariant set $\mathcal{A}\!\uparrow$. The two remaining possibilities are, in fact, the two maximally permissive controls for this state, namely, F1 and P6 can be enabled, or F5 and P4 can be enabled.

6. CONCLUSION

Controlled Petri nets provide a modeling formalism for controlled discrete event processes which admits the formulation of control synthesis problems. Other models proposed for such processes provide the same advantage. The primary motivation for introducing CPNs is to take advantage of the qualitative and quantitative analysis tools that exist for standard Petri nets. Application of these tools in the domain of controlled systems is an open area for future research. Promising directions for further developments include: relationships between Petri net invariants control invariants in CPNs; development of dynamic feedback logic using CPNs as controllers; synthesis of feedback logic under partial state information; introduction of timing and performance criteria for selecting controls.

ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under research grant number DMC-8451493.

REFERENCES

1. J.L. Peterson, *Petri Net Theory and the Modeling of Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1981.
2. P.J. Ramadge and W.M. Wonham, "Modular Feedback Logic", Tech. report Systems Control Group Rpt. No. 8511, University of Toronto, August 1985, also, *SIAM J. Control and Optimization*, vol. 25, no. 5, Sept. 1987.

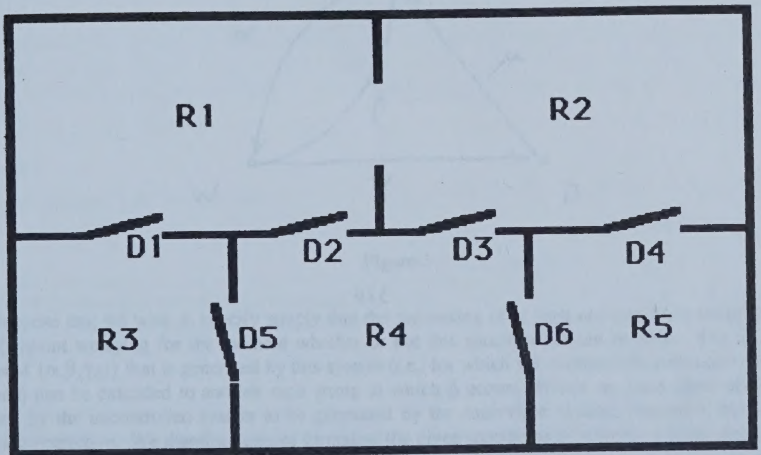


Figure 1: Cat-mouse maze

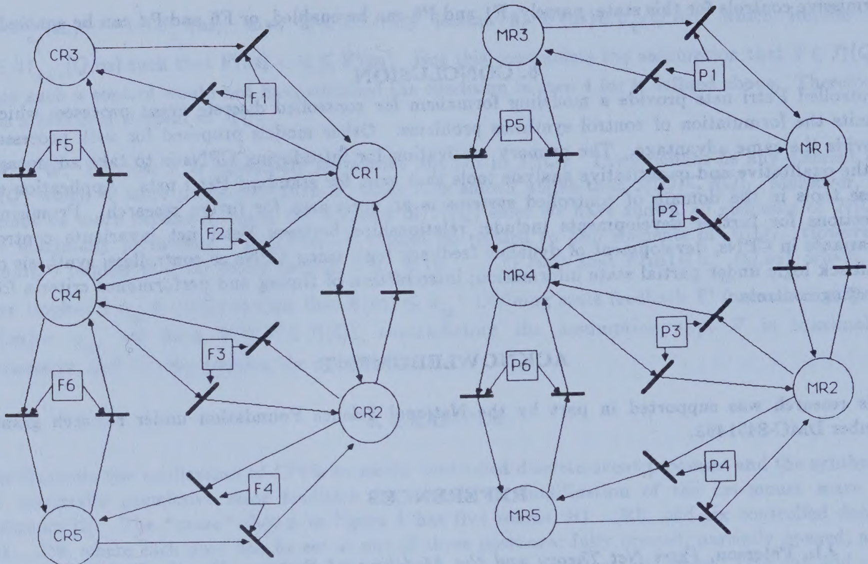

CR3
F1
F5
CR1
F2
CR4
F3
F6
CR2
F4
CR5
MR3
P1
P5
MR1
P2
MR4
P3
P6
MR2
P4
MR5

Figure 2: CPN model of cat and mouse locations

SUPERVISORY CONTROL WITH INFINITE-STRING SPECIFICATIONS

J.G. THISTLE and W.M. WONHAM
Department of Electrical Engineering
University of Toronto
Toronto, Ontario
CANADA M5S 1A4

ABSTRACT

Formal languages consisting of infinite strings, or ω-*languages*, allow the specification of eventualities, which ordinary formal languages do not. Some fundamental properties of ω-language specifications for supervisory control are defined. A supervisor synthesis problem involving ω-language specifications is posed, and necessary and sufficient conditions for its solvability are given.

1. INTRODUCTION

This paper deals with the use of ω-*languages* [1,2], or formal languages consisting of infinite strings, in specifications for discrete event systems. Ramadge and Wonham [3] and others have used ordinary formal languages (consisting of finite strings) to specify discrete event system behavior. A drawback of this approach is that it does not allow us to state that some event must eventually occur, or that it must occur infinitely often -- in the language of program verification, it does not allow the specification of "eventualities". Consider for instance the following simple system:

Example 1.1

The transition diagram in figure 1 represents a machine in an automated factory. Its states are idle (I), working (W) and down (D). The transition labelled by α represents the commencement of work on a part and that labelled by β represents its completion. The transition γ is a breakdown of the machine and μ is its repair.

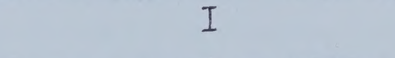


Figure 1

Suppose that we wish to specify simply that the processing of at least one part must eventually be completed (without worrying for the moment whether or not this specification can be met). Any finite string of symbols of $\{\alpha,\beta,\gamma,\mu\}$ that is generated by this system (i.e., for which the corresponding sequence of transitions is defined) can be extended to another such string in which β occurs. Hence we must allow all finite strings generated by the uncontrolled system to be generated by the supervised system; otherwise, our specification will be too restrictive. We therefore cannot formalize the given specification in terms of finite strings.

The problem with the finite string approach is that it deals only with finite prefixes of potentially infinite strings. If we instead refer to the set of infinite strings to be generated by the supervised system, the specification can be formalized easily: this set must be contained in

$$(\alpha\gamma\mu)^* \alpha\beta[\alpha(\beta+\gamma\mu)]^{\omega}$$

(In our examples we assume that the reader is familiar with regular expressions [4]. We augment this notation by using terms like $[\alpha(\beta+\gamma\mu)]^\omega$, which denotes the set of infinite strings made up of strings in $\alpha(\beta+\gamma\mu)$.) In fact if we instead use the language

$$[(\alpha\gamma\mu)^*\alpha\beta]^\omega ,$$

we can specify that β occur infinitely often. Thus, ω-languages are an attractive means of specification. The present paper contains some basic results on the use of ω-language specifications in discrete event systems control.

In section 2 we present some formal language preliminaries. Section 3 covers the formal definitions of discrete event systems and their supervisors. In section 4 we define the fundamental properties of Σ_u-controllability and realizability, which determine whether a given ω-language can be "synthesized" by means of supervisory control. We define in section 5 a supervisor synthesis problem based on ω-language specifications and state a necessary and sufficient condition for its solvability. This involves an additional property called Σ^ω-controllability. Finally, in section 6 we give some concluding remarks on our results and on the relationship of our work to that of others. Proofs are presented in section 7.

2. FORMAL LANGUAGE PRELIMINARIES

Let Σ be an *alphabet* of symbols. Σ^* denotes the set of all finite strings over Σ, including the empty string 1. Any subset $L\subset\Sigma^*$ is a *language* over Σ. If $s,t\in\Sigma^*$, then st represents the catenation of s and t. If $L_1,L_2\subset\Sigma^*$, then L_1L_2 is the set of catenations st, where $s\in L_1$ and $t\in L_2$.

Σ^ω denotes the set of infinite strings over Σ, and any subset $L\subset\Sigma^\omega$ is an *ω–language* over Σ. If $L\subset\Sigma^*\cup\Sigma^\omega$, then $P(L)$ is the set of finite prefixes of L. For any $L\subset\Sigma^*$, $\bar{L} := \{s\in\Sigma^\omega : P(\{s\})\subset P(L)\}$. (In Boasson and Nivat's nomenclature, $\bar{L}$ is the "adherence" of L [2].)

3. SUPERVISORY CONTROL

Following Ramadge and Wonham [3], we model the discrete event system (DES) that is to be controlled as an automaton

$$\mathbf{G} = (\Sigma,Q,\delta,q_0) ,$$

where Σ is an alphabet of *event symbols*, Q is the set of *states*, $q_0\in Q$ is an *initial state* and

$$\delta : \Sigma\times Q \to Q$$

is a (partial) *transition function.*

We extend the map δ in the standard fashion:

$$\delta : \Sigma^*\times Q \to Q$$

is defined by

$$\delta(1,q) = q$$

&

$$\delta(s\sigma,q) = \delta(\sigma,\delta(s,q)), \ \forall s\in\Sigma^*,\sigma\in\Sigma .$$

The language *generated* by $\mathbf{G}$ is

$$L(\mathbf{G}) := \{s : s\in\Sigma^* \ \& \ \delta(s,q_0) \text{ is defined}\} .$$

The language $L(\mathbf{G})$ represents the set of finite trajectories of the system -- the finite sequences of events that are physically possible. The set of infinite trajectories is represented by the ω-language *generated* by $\mathbf{G}$:

$$L_\omega(\mathbf{G}) := \{s\in\Sigma^\omega : \exists q_0,q_1,q_2,\ldots\in Q \ s.t. \ (\forall_i>0)\delta(\sigma_i,q_{i-1}) = q_i , \quad \text{where } \sigma_i \text{ is the } i^{th} \text{ symbol in } s\} .$$

(Though we have given a deterministic model here, our results also hold for systems with "bounded nondeterminism" -- where $\delta : \Sigma\times X\to\{X'\subset X : |X'|<\infty\}$ - when $\mathbf{G}$, $L(\mathbf{G})$ and $\mathbf{G}$ are defined in the natural way.)

We introduce a control feature by identifying a subset of *controllable* events $\Sigma_c\subset\Sigma$, the remaining events comprising the set of *uncontrollable* events, $\Sigma_u := \Sigma - \Sigma_c$. Intuitively, controllable events can be disabled, or prevented from occurring, while uncontrollable events cannot.

The control action is performed by a *supervisor:*

$$\mathbf{S} = (S,\phi) ,$$

where $S = (\Sigma, X, \xi, x_0)$ is an automaton and $\phi : \Sigma \times X \rightarrow \{0,1\}$ is a total function called the *feedback map*. This map represents the mechanism for disabling controllable events.

Accordingly, we define the *controlled discrete event system* (CDES) to be the automaton

$$S/G := (\Sigma, X \times Q, (\xi \times \delta)^{\phi}, (x_0, q_o)) ,$$

where $(\xi \times \delta)^{\phi} : \Sigma \times X \times Q \rightarrow X \times Q$ is defined according to

$$(\xi \times \delta)^{\phi}(\sigma, x, q) = \begin{cases} (\xi(\sigma,x), \delta(\sigma,q)) & \text{if } \phi(\sigma,x)=1 \\ \text{undefined} & \text{if } \phi(\sigma,x)=0 \end{cases}$$

As is noted in [3], this definition is physically meaningful only if S is a *complete* supervisor for G; i.e. only if the three conditions

(i) $s \in L(S/G)$

(ii) $s\sigma \in L(G)$ (i.e.$\delta(\sigma, \delta(s, x_0))$ is defined)

(iii) $\phi(\sigma, \xi(s, x_0))=1$

together imply that

(iv) $\xi(\sigma, \xi(s, x_0))$ is defined.

We call $L(S/G)$ (resp. $L^{\omega}(S/G)$) the language (resp. ω-language) *synthesized* by S.

We say that a supervisor S for G is *nonblocking* if

$$L(S/G) = P(L^{\omega}(S/G)) ;$$

that is, if every finite string generated by the controlled system can be extended to an infinite one. (Thus, nonblocking supervisors yield "deadlock-free" systems.) In this paper we shall concern ourselves only with nonblocking supervisors.

4. Σ_u- CONTROLLABILITY AND REALIZABILITY

This section deals with the two properties that determine whether a given ω-language can be synthesized by means of a complete, nonblocking supervisor.

Definition 4.1:

Suppose that a plant G is given. An ω-language $L \subset L^{\omega}(G)$ is *Σ_u-controllable* (w.r.t. G) if $P(L)\Sigma_u \cap L(G) \subset P(L)$.

Σ_u-controllability is very similar to Ramadge and Wonham's notion of controllability. Indeed, $L \subset \Sigma^{\omega}$ is Σ_u-controllable if and only if $P(L)$ is controllable according to [3]. Roughly speaking, a Σ_u-controllable language represents a specification of system behavior that makes allowances for the occurrence of uncontrollable events.

Definition 4.2:

An ω-language $L \subset \Sigma^{\omega}$ is *realizable* if

$$L = \overline{P(L)} .$$

A realizable language L is one that includes all strings $s \in \Sigma^{\omega}$ such that all prefixes of s are prefixes of strings in L. As we shall see below, any language generated by a CDES is realizable, and this reflects the fact that a CDES is causal and deterministic : events are disabled solely on the basis of past events, so if the CDES generates all of the finite prefixes of an infinite string, it must also generate the entire string. The property of realizability of ω-languages is related to that of continuity of functions in Scott's theory of computation [5].

Note that eventuality specifications like that in example 1.1 correspond to ω-languages that are not realizable. This is precisely because they impose no restriction on the set of infinite strings generated by the system but they do restrict the set of infinite strings generated.

Proposition 4.3:

Let G be a DES. For any $L \subset L^{\omega}(G)$, there exists a complete, nonblocking supervisor S for G that synthesizes L if and only if L is Σ_u-controllable and realizable.

5. A SYNTHESIS PROBLEM

In this section we give a necessary and sufficient condition for the solvability of a supervisor synthesis problem. As in [3] and subsequent papers, we assume that two specification languages are given, one representing the "maximal legal behavior," and the other the "minimal acceptable behavior" of the system. This leads to the following problem statement:

Synthesis Problem (SP)

Given a DES G and two ω-languages A and E such that $A \subset E \subset L^{\omega}(G)$, construct a complete nonblocking supervisor S for G such that

$$A \subset L^{\omega}(S/G) \subset E \, .$$

By proposition 4.3, $L^{\omega}(S/G)$ is Σ-controllable and realizable for any CDES S/G. We therefore define the following language classes:

Definition 5.1:

Suppose that the DES G is given. For any $L \subset L^{\omega}(G)$, $C_u(L) := \{L' \subset L : L' \text{ is } \Sigma_u\text{-controllable w.r.t. } G\}$.
For any $L \subset \Sigma^{\omega}$, $R(L) := \{L' \subset \Sigma^{\omega} : L' \supset L \text{ and } L' \text{is realizable}\}$.

Proposition 5.2:

(i) The language class $C_u(L)$ contains a supremal element ("sup $C_u(L)$").

(ii) The language class $R(L)$ contains an infimal element, ("inf $R(L)$"); namely, $\overline{P(L)}$.

By proposition 4.3, it is clear that a necessary condition for the solvability of SP is that inf $R(A)$ be contained in sup $C_u(E)$. Unfortunately, this condition is not sufficient, as the following example shows.

Example 5.3:

Consider once again the system described in the example 1.1. Let $\Sigma_c = \{\alpha,\beta,\mu\}$ and $\Sigma_u = \{\gamma\}$. For specifications, take

$$A = (\alpha\beta)^{\omega} \text{ and } E = [\alpha(\beta+\gamma\mu)]^{*}(\alpha\beta)^{\omega} \, .$$

Then $A = \inf R(A) \subset \sup C_u(E) = E$, but in order to satisfy $L^{\omega}(S/G) \subset E$ the supervisor S would have to ensure that the uncontrollable event γ eventually stopped occurring. This is clearly impossible.

The existence of a nonblocking supervisor requires that the plant dynamics possess special global structure which is not captured by the condition inf $R(A) \subset$ sup $C_u(E)$. In order to describe this special structure it is convenient to introduce another notion of controllability:

Definition 5.4:

(i) For any $L \subset \Sigma^{\omega}$ and any nonempty $K \subset P(L)$, K is an *L-keeper* if

$$(\forall s \in K)[(\exists\sigma\in\Sigma)(s\sigma\in P(L)-K) \rightarrow (\exists\sigma'\in\Sigma_u)(s\sigma'\in K)]$$

(L-keepers are subsets of $P(L)$ in which the system's trajectory may become "trapped" due to the occurrence of uncontrollable events.)

(ii) For any $L \subset \Sigma^{\omega}$ and any $K \subset P(L)$, K is *L-transient* if there is no $s \in K$ such that all sufficiently long prefixes of s belong to K.

(K is L-transient if there is no trajectory in L that eventually enters K and remains there.)

(iii) A language $L \subset \Sigma^{\omega}$ is *Σ^{ω}-controllable* if $P(L)$ contains no L-transient L-keepers.

(Thus a Σ^{ω}-controllable language allows for the possibility of the system's trajectory being "trapped" in L-keepers.)

Definition 5.5:

Suppose that a DES G is given. Then for any $L \subset L^{\omega}(\mathrm{G})$,

$$C(L) := \{L' \subset L : L' \text{ is } \Sigma_u\text{– and } \Sigma^{\omega}\text{–controllable}\} .$$

Proposition 5.6:

For any DES G and any ω-language $L \subset L^{\omega}(\mathrm{G})$, the class $C(L)$ contains a supremal element ("sup $C(L)$").

Note that if $L \subset \Sigma^{\omega}$ is realizable, then $P(L)$ contains no L-transient subsets. Hence every realizable ω-language is Σ^{ω}-controllable.

Theorem 5.7:

SP is solvable if and only if $\inf R(A) \subset \sup C(E)$.

Example 5.8

Returning to example 4.3, we see that E is not Σ^{ω}-controllable, since it contains the E-transient E-keeper $(\alpha\gamma\mu)^*$. In fact, $\sup C(E) = \phi$, so the specification cannot be satisfied.

6. CONCLUSION

We have defined the properties of realizability, Σ_u-controllability and Σ^{ω}-controllability of ω-language specifications for discrete event systems. These are of interest because nonrealizable ω-languages can express specifications that ordinary formal languages cannot. We have also given a necessary and sufficient condition for the solvability of a supervisor synthesis problem involving ω-language specifications.

The authors know of little other work on the use of ω-languages in control theory. In [6], Arnold and Nivat consider only systems for which, in our terminology, all events are controllable. Their results characterize the relationships between behaviors achieved by using different control actions (i.e. feedback maps) with a fixed controller state structure. Fusaoka et al [7] give an example of the synthesis of a feedback map from an ω-automaton. The ω-automaton (a machine that "recognizes" an ω-language [1, 7-10]) is derived from a temporal logic specification, essentially by means of Wolper's tableau mehtod [10].

The results given in the present paper will be of use in extending the method of Fusaoka et al to the synthesis of dynamic controllers. This is a current topic of research.

7. PROOFS

The following lemma will be useful:

Lemma 7.1:

For any (C)DES G,

$$L^{\omega}(\mathrm{G}) = \overline{L(\mathrm{G})}$$

Proof:

By definition of $L^{\omega}(\mathrm{G})$

□

Proof of Proposition 4.3:

(if) By Σ_u-controllability, $P(L)$ is controllable in the sense of [3]. It follows that there exists a complete supervisor S for G such that

$$L(\mathrm{S/G}) = P(L) .$$

By lemma 7.1,

$$L^{\omega}(\mathrm{S/G}) = \overline{P(L)} = L\ .$$

(only if)

$$P(L^{\omega}(\mathrm{S/G})) \subset L(\mathrm{S/G})$$
$$\Rightarrow \overline{P(L^{\omega}(\mathrm{S/G}))} \subset \overline{L(\mathrm{S/G})}$$
$$= L^{\omega}(\mathrm{S/G}) \qquad \text{(by lemma 7.1)}$$

Hence $L^{\omega}(\mathrm{S/G})$ is realizable.

For Σ_u-controllability of $L^{\omega}(\mathrm{S/G})$, suppose that $\mathrm{S} = (s,\phi)$, where $s = (\Sigma, X, \xi, x_0)$. Let $s \in P(L^{\omega}(\mathrm{S/G}))$, $\sigma \in \Sigma_u$ and so $\in L(\mathrm{G})$. Then

$$\phi(\sigma, \xi(s,x_0)) = 1 \qquad (\sigma \in \Sigma_u)$$

so

$$\xi(\sigma, \xi(s,x_0)) = 1 \qquad \text{(by completeness of S)}$$

Hence

$$P(L^{\omega}(\mathrm{S/G}))\Sigma_u \cap L(\mathrm{G}) \subset L(\mathrm{S/G})$$
$$= P(L^{\omega}(\mathrm{S/G})) \qquad (s \text{ is nonblocking})$$

□

Proof of Proposition 5.2:

(i) The empty language belongs to $C_u(L)$, so $C_u(L)$ is nonempty. Moreover $C_u(L)$ is closed under arbitrary unions.

(ii) Clearly, $\overline{P(L)} \in R(L)$. Suppose that there exists $K \in R(L)$ such that

$$L \subset K \subset \overline{P(L)}$$
$$\Rightarrow P(L) \subset P(K) \subset P(L)$$
$$\Leftrightarrow K = \overline{P(K)} = \overline{P(L)}$$

□

Proof Outline for Proposition 5.6:

Define the operator $\Omega : 2^{\Sigma^{\omega}} \rightarrow 2^{\Sigma^{\omega}}$ by

$$\Omega(L) := \{s \in L : (\forall K \subset P(L))\ [K \text{ is an } L\text{–transient } L\text{–keeper} \Rightarrow P(\{s\}) \cap K = \phi]\}\ .$$

(Thus, $\Omega(L)$ is the set of all strings in L that do not "visit" L-transient L-keepers.) This operator has the following properties:

(i) $\Omega(L)$ is Σ^{ω}-controllable.

(ii) Ω preserves Σ_u-controllability.

(iii) If $L_1 \subset L_2 \subset \Sigma^{\omega}$ and L_1 is Σ_u-controllable, then $\Omega(L_1) \subset \Omega(L_2)$.

It follows that $\Omega(\sup C_u(L))$ is the supremal element of $C(L)$.

□

Proof Outline for Theorem 5.7:

(only if)

It is easy to see that every realizable language is Σ^{ω}-controllable; the result follows by by proposition 4.3.

(if)

Suppose that

$$A' := \inf R(A) \subset \sup C(E) =: E'\ .$$

Consider the infinite tree whose nodes are the elements of $P(E')$ and where, for any node s, the children of s are exactly those strings $s\sigma \in P(E')$, where $\sigma \in \Sigma$. The proof is best explained by means of an informal argument about "pruning" this tree.

To each branch in the tree there corresponds in an obvious way a string in $\overline{P(E')}$. It suffices to show therefore that the tree can be pruned in such a way that the ω-language L corresponding to the set of infinite branches in the pruned tree is realizable and Σ_u-controllable, and that $A' \subset L \subset E'$.

Now L will automatically be realizable (by definition). And since E' is Σ_u-controllable, we can ensure that L is Σ_u-controllable by deleting only nodes $s\sigma$ where $\sigma \in \Sigma_c$ -- and only doing so if there is a node $s\sigma'$ in the tree, where $\sigma' \neq \sigma$. We will have $A' \subset L$ as long as we delete no nodes in $P(A')$. Finally, we can obtain $L \subset E'$ if we make certain that in our pruning we cut every infinite branch that represents a string in $\overline{P(E')} - E'$.

These branches have two important properties which make it possible to prune the tree appropriately: first, since A' is realizable and $A' \subset E'$,

(a) each such branch contains only finitely many nodes in $P(A')$;

and second, since E' is Σ^ω-controllable,

(b) each such branch contains infinitely many nodes $s\sigma$ such that $\sigma \in \Sigma_c$ and there exists a node $s\sigma'$ in the tree, for some $\sigma' \neq \sigma$.

These facts allow us to prune the tree by performing the following steps:

(i) For every node $s\sigma$ such that $\sigma \in \Sigma_u$, eliminate every node $s\sigma' \notin P(A')$ such that $\sigma' \in \Sigma_c$. Note that this preserves (b).

Assume without loss of generality that there are only two controllable events, α and β. For every node $s\alpha \notin P(A')$ in a given branch, call s a β-*node* of the branch if there exists a node $s\beta$ in the tree. Define α-nodes similarly.

(ii) For every branch that corresponds to a string in $\overline{P(E')} - E'$ and contains only an *odd* finite number of β-nodes, delete the node $s\alpha$, where s is the deepest (i.e. furthest from the root) of all the β-nodes in the branch. This does not result in any other branches having only a finite number of β-nodes, since we eliminate only nodes of the form $s\alpha$. It is not hard to see, by parity considerations, that (b) is preserved.

(iii) For every branch that corresponds to a string in $\overline{P(E')} - E'$ and contains only an *even* finite number of β-nodes (or none), delete the node $s\beta$, where s is the shallowest α-node that is deeper than all the β-nodes in the branch (if any exist). We leave it to the reader to show, by parity considerations, that this does not result in any other branches in the tree having only a finite number of β-nodes.

(iv) At this stage, every remaining branch that corresponds to a string in $\overline{P(E')} - E'$ contains β-nodes (in fact, infinitely many of them). We can therefore eliminate all such branches by deleting all nodes $s\alpha$ in the tree, where s is a β-node.

□

8. REFERENCES

[1] Cohen, Rina S., and Arie Y. Gold, 1977. 'Theory of ω-languages', parts I and II, Journal of Computer and System Sciences **15**, pp. 169-208.

[2] Boasson, Luc, and Maurice Nivat, 1980. 'Adherences of languages', Journal of Computer and System Sciences **10**, pp. 285-309.

[3] Ramadge, P.J., and W.M. Wonham, 1983. 'Supervisory control of a class of discrete event processes', Systems Control Group Report #8311, Dept. of Electrical Engineering, University of Toronto. Also, SIAM J. Control and Optimization **25** (1), 1987, pp. 206-230.

[4] Hopcroft, J.E., and Ullman, J.D., 1979. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Pub. Co., Reading.

[5] Scott, Dana, 1970. 'Outline of a mathematical theory of computation', Technical Monograph PRG-2, Oxford University Computing Laboratory Programming Research Group. Also, Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems, Princeton University, 1970, pp. 169-176.

[6] Arnold, A., and M. Nivat, 1980. 'Controlling behaviors of systems: some basic concepts and some applications', *Mathematical Foundations of Computer Science 1980*, Lecture Notes in Computer Science #88, Springer-Verlag, Berlin.

[7] Fusaoka, Akira, Hirohisa Seki and Kazuko Takahashi, 1983. 'A description and reasoning of plant controllers in temporal logic', Proc. 8th International Joint Conference on Artificial Intelligence, Karlsruhe, W. Germany, pp. 405-408.

[8] Eilenberg, Samuel, 1974. *Automata, Languages, and Machines,* vol. A, Academic Press, New York.

[9] McNaughton, Robert, 1966. 'Testing and generating infinite sequences by a finite automaton', Information and Control **9**, pp. 521-530.

[10] Wolper, Pierre Léon, 1982. 'Synthesis of communicating processes from temporal logic specifications', Rpt. No. STAN-CS-82-925, Dept. of Computer Science, Stanford University. Also (with Z. Manna), ACM Trans. on Programming Languages and Systems **6** (1), 1984, pp. 68-93.

DECIDABILITY FOR TEMPORAL LOGIC IN CONTROL THEORY

J. F. KNIGHT[1]
Dept. of Mathematics
Univ. of Notre Dame
Notre Dame, IN 46556

K. M. PASSINO[2]
Dept. Electrical & Computer Eng.
Univ. of Notre Dame
Notre Dame, IN 46556

ABSTRACT

Some temporal logics are known to be decidable; i.e., there is a mechanical procedure which, when applied to any formula in the temporal language, will terminate in a finite number of steps and say whether the formula is necessarily true or not. Here we develop a temporal logic in a control theoretic framework. We show for the type of plant for which our language is appropriate, while the behavior need not ever become periodic, we could assume that it does. From this, we prove results on decidability, which are used in control system analysis and controller synthesis. The results are illustrated on two examples.

§ 1. Introduction

We imagine a plant in which the information about current conditions and the mechanisms for control are limited, and there are significant unpredictable, uncontrollable forces at work. We assume that the sensor information at a given time can be coded by a symbol from a finite alphabet, and there are finitely many possible states for the controller. As a simple example, consider a surge tank with sensors distinguishing among five liquid levels (empty, low, normal, high, and full) and with the controller simply opening and closing a fill valve. Unpredictable users operate another valve to take liquid from the tank.

We use a propositional temporal language to state closed loop specifications. The propositional variables are members of either the set **Q** of controller states or the sensor alphabet **A**. There are the usual propositional connectives. We use three temporal (modal) operators: O (next), □ (for all times now and later), and ◊ (for some time now or later). We think of time as being discrete. The precise meaning of O depends on the controller's schedule. It may be that once every second the controller looks at the sensor symbol and responds. Then O means "next second". Or, it may be that the controller is alerted when the sensor symbol changes and responds then. In this case, O means "next time something happens".

The running of the closed loop system gives rise to an infinite string of sensor symbols and a corresponding sequence of controller states. The controller acts as a deterministic finite-state machine. A controller strategy consists of an initial state $q \in \mathbf{Q}$ and a transition function $t:\mathbf{Q}x\mathbf{A}\rightarrow\mathbf{Q}$. If the current state is s and the symbol a is being sensed, then the controller goes to state $s'=t(s,a)$. The controller has some influence over the string of symbols, but not full control. The response of the plant to the controller is non-deterministic. We have a non-empty set **I**, where $\mathbf{A}\supseteq\mathbf{I}$, of possible initial symbols, and a relation N which associates with each pair $(a,s)\in \mathbf{A}x\mathbf{Q}$ (where s is the state that resulted from sensing a) a non-empty set **A'** of possible next symbols from **A** .

In the example of the surge tank, let symbols a_1, a_2, a_3, a_4, and a_5 denote the levels (from lowest to highest), and let the states be q_0 (fill valve closed) and q_1 (fill valve open). Suppose that the controller is responding once a second. It may be that if the current symbol is a_4 (the level is high but not full) and the controller's response to that has been q_0, then the next symbol must be either a_4 or a_3 (because even if the user valve stays open for a full second, the level cannot drop to low). This is the sort of information carried in N.

For a particular plant, with a fixed **I** and N, there are finitely many possible controller strategies (q,t), and for each (q,t), there may be continuum many strings of sensor symbols α that satisfy all of the constraints given by **I**, N, q, and t. These are the (q,t)-allowable strings. We shall define satisfaction of a formula ϕ by a triple (q,t,α), where (q,t) is a controller strategy and α is a (q,t)-allowable string. We show that for any (q,t)-allowable string α, there is an "ultimately periodic" string β such that (q,t,α) satisfies ϕ if an only if (q,t,β) does. We say that the strategy (q,t) makes the formula ϕ valid if ϕ is satisfied by (q,t,α) for all (q,t)-allowable strings. We

---

[1] Partially supported by National Science Foundation Grants DMS 85 03353 and DMS 87 01559

[2] Partially supported by Univ. of Notre Dame, Jesse Jones Research Grant

show that there is a mechanical procedure for deciding in a finite number of steps whether a given controller strategy (q,t) makes a given closed loop specification formula ϕ valid. From this, it will follow that for a given formula ϕ, we can effectively either find a controller strategy (q,t) making ϕ valid, or else say for sure that no such strategy exists. The decision procedure is not computationally practical, in general. A tableau method would be more efficient. However, most specifications of real interest seem to have a simple form (low "rank"), and if we did a certain part of the decision procedure for a given plant, we could then examine a whole family of low rank specifications easily.

The propositional temporal language we use is like that in Manna and Wolper [1] except that we do not include the "until" operator. Manna and Wolper obtain decidability and synthesis results using the method of semantic tableaus. Our proof is closer in spirit to work of Buchi [2]. Thistle and Wonham [3] use a language with a slightly different syntax. They also allow a more general sort of plant and controller than we do (e.g., the set of controller states may be infinite). They do not prove decidability. Instead, they describe a system of rules of proof, and illustrate its use by proving closed loop specification formulas from axioms for the plant and controller in various specific examples. Our work is also related to recent work of Ostroff [4]. Ostroff significantly extended Thistle and Wonham's work, and obtained computationally practical decision procedures for a special class of formulas in his language. In our treatment, the plant specifications and the controller strategy are incorporated in a natural way into the semantics of the language. Our decidability proof seems to be new. It is completely elementary and self-contained.

In Section 2, we describe the syntax and semantics of our language, and define a notion of rank which quantifies the complexity of a temporal formula. In Section 3, we give the decidability and synthesis results. Section 4 contains some examples of control system analysis and synthesis, illustrating the various notions and results from the earlier sections.

§ 2. The Formal Language

In this section, we describe precisely the syntax and semantics of our language. We first say what the symbols are. The set of propositional variables is $\mathbf{A}\cup\mathbf{Q}$, where $\mathbf{A}$ is the sensor alphabet and $\mathbf{Q}$ is the set of controller states. The temporal operators are O, $\Box$, and $\Diamond$. The logical connectives are the usual ones $\neg$ (not), & (and), $\vee$ (or), $\rightarrow$ (implies), and $\leftrightarrow$ (if and only if). Finally, there are the parentheses (, and).

The rules for forming formulas are as follows:

(1) A single propositional variable a for $a\in\mathbf{A}$ or q for $q\in\mathbf{Q}$ is a formula.

(2) If ϕ is a formula, then so is $\neg\phi$.

(3) If ϕ and ψ are formulas, then so are $(\phi\&\psi)$, $(\phi\vee\psi)$, $(\phi\rightarrow\psi)$,and $(\phi\leftrightarrow\psi)$.

(4) If ϕ is a formula, then so are $O\phi$, $\Box\phi$, and $\Diamond\phi$.

(5) Nothing is a formula unless it can be obtained by finitely many applications of (1)-(4) above.

Here are some sample formulas of the language for the example of the surge tank. Recall that the sensor symbols $a_1, \ldots, a_5$ represent levels of liquid in the tank (from empty to full), and the states q_0 and q_1 indicate that the fill valve is closed, open, respectively.

1. $\Box\Diamond a_3$
2. $\Box((a_2 \vee a_4) \rightarrow Oa_3)$
3. $(\Box(a_1 \rightarrow Oq_1) \rightarrow \Box(a_1 \rightarrow \Diamond a_2))$

Formula 1 says that the level is infinitely often normal. Formula 2 says that any time the level is low or high, it will next be normal. Formula 3 says that if the fill valve opens whenever the tank is empty, then each time the tank becomes empty it will eventually reach low again.

We next define the rank of a formula ϕ, denoted by $r(\phi)$. The definition proceeds by induction on ϕ.

(a) $r(\phi)=0$ if ϕ consists of a single propositional variable,

(b) $r(\neg\phi)=r(\phi)$,

(c) $r((\phi\&\psi))=r((\phi\vee\psi))=r((\phi\rightarrow\psi))=r((\phi\leftrightarrow\psi))=\sup\{r(\phi),r(\psi)\}$,

(d) $r(O\phi)=r(\Box\phi) = r(\Diamond\phi)=r(\phi)+1$.

The three sample formulas given above all have rank 2, and, in fact, the specifications that we have thought of in various natural examples all seem to have rank 1 or 2. There is no "prenex normal form" for temporal logic. In fact, what we have expressed by Formula 2 above could not be expressed by any formula in which all the temporal operators occur at the front. The rank measures the complexity of a formula by counting the number of nested temporal operators without bringing them to the front. Rank does not go up when we form complicated Boolean combinations (putting formulas together with $\neg$, &, $\vee$, $\rightarrow$, and $\leftrightarrow$). The formulas of rank 0 are the Boolean combinations of the basic formulas from the set $B_0=\{p: p\in A\cup Q\}$. If $n>0$ and B_n is the set of basic rank n formulas of the forms $O\phi$, $\Box\phi$, and $\Diamond\phi$, where ϕ has rank n-1, then all formulas of rank n are obtained as Boolean combinations of formulas from B_n and formulas of lower rank.

We have described the syntax of the language. We now turn to the semantics. The set I, of initial sensor symbols, and the relation N, giving possible next sensor symbols, are determined by the plant. For any set X, let X^ω denote the set of all infinite strings of elements of X. Let (q,t) be a controller strategy and let $\alpha\in A^\omega$. We say that α is (q,t)-allowable if there is a string $\sigma\in Q^\omega$ such that the following conditions are all satisfied:

(1) $\alpha_0\in I$,

(2) $\sigma_0=q$,

(3) for all i, $t(\sigma_i,\alpha_i)=\sigma_{i+1}$,

(4) for all i, $(\alpha_i,\sigma_{i+1},\alpha_{i+1})\in N$.

If α is a (q,t)-allowable string, let $q^{\alpha,t,i}$ be the state reached after i steps when the controller strategy (q,t) is applied to α (this is the state denoted by σ_i in the definition above). Let α^i be the result of dropping the first i terms from α.

We are now ready to define satisfaction. If (q,t) is a controller strategy and α is a (q,t)-allowable string, we shall use the notation $(q,t,\alpha)\models\phi$ to indicate that the formula ϕ is satisfied by (q,t,α). The definition proceeds by induction on ϕ.

(1) $(q,t,\alpha)\models p$ if $p\in Q$ and $p=q$ (the initial state) or if $p\in A$ and $p=\alpha_0$,

(2) $(q,t,\alpha)\models\neg\phi$ if it is not the case that $(q,t,\alpha)\models\phi$,

(3) $(q,t,\alpha)\models(\phi\&\psi)$ if $(q,t,\alpha)\models\phi$ and $(q,t,\alpha)\models\psi$,

(4) $(q,t,\alpha)\models(\phi\vee\psi)$ if $(q,t,\alpha)\models\phi$ or $(q,t,\alpha)\models\psi$,

(5) $(q,t,\alpha)\models(\phi\rightarrow\psi)$ if $(q,t,\alpha)\models\phi$ implies $(q,t,\alpha)\models\psi$,

(6) $(q,t,\alpha)\models(\phi\leftrightarrow\psi)$ if $(q,t,\alpha)\models\phi$ if and only if $(q,t,\alpha)\models\psi$,

(7) $(q,t,\alpha)\models O\phi$ if $(q^{\alpha,t,1},t,\alpha^1)\models\phi$,

(8) $(q,t,\alpha)\models\Box\phi$ if for all $i\geq 0$, $(q^{\alpha,t,i},t,\alpha^i)\models\phi$,

(9) $(q,t,\alpha)\models\Diamond\phi$ if for some $i\geq 0$, $(q^{\alpha,t,i},t,\alpha^i)\models\phi$.

An equivalence relation on a set X is a two place relation ~ (on X) that is reflexive, symmetric, and transitive. If $x\in X$, then the ~-equivalence class of x is the set $\{y\in X: y\sim x\}$. We may speak of ~-equivalence classes in general (without naming elements). We next define equivalence relations $\sim_{n,(q,t)}$, on the set of (q,t)-allowable strings, such that equivalence implies satisfaction of the same formulas of rank $\leq$ n. Suppose that α and β are (q,t)-allowable strings of symbols.

(a) $\alpha\sim_{0,(q,t)}\beta$ if $\alpha_0=\beta_0$

(b) $\alpha\sim_{n+1,(q,t)}\beta$ if (i) $\alpha\sim_{n,(q,t)}\beta$, (ii) $q^{\alpha,t,1}=q^{\beta,t,1}$ and $\alpha^1\sim_{n,(s,t)}\beta^1$, where $s=q^{\alpha,t,1}$, and (iii) for all i there exists a j (and for all j there exists an i) such that $q^{\alpha,t,i}=q^{\beta,t,j}$ and $\alpha^i\sim_{n,(s,t)}\beta^j$ where $s=q^{\alpha,t,i}$.

Note that $\alpha\sim_{0,(q,t)}\beta$ iff α and β have the same first symbol; $\alpha\sim_{1,(q,t)}\beta$ iff α and β have the same first symbol and the same second symbol, and the set of pairs representing a current state and symbol being read in that state is the same when the controller strategy (q,t) is applied to α and to β. For example, if $\alpha=a_2a_1a_1a_2a_3a_3a_3\ldots$, and $\beta=a_2a_1a_2a_3a_3a_3\ldots$, and (q,t) is a controller strategy such that $t(q,a_i)=q$ for i=1,2,3, then $\alpha\sim_{0,(q,t)}\beta$ and $\alpha\sim_{1,(q,t)}\beta$, but not $\alpha\sim_{2,(q,t)}\beta$. It is easy to see that if $m<n$ and $\alpha\sim_{n,(q,t)}\beta$, then $\alpha\sim_{m,(q,t)}\beta$. We shall make use of this fact later.

§ 3. Decidability and Synthesis Results

The decidability and synthesis results are given in this section. The proofs are based on three lemmas. The first lemma says that strings which are $\sim_{n,(q,t)}$-equivalent will satisfy the same formulas of rank n.

Lemma 1. Let ϕ be a formula such that $r(\phi)\leq n$, and let α and β be (q,t)-allowable strings such that $\alpha\sim_{n,(q,t)}\beta$. Then $(q,t,\alpha)|=\phi$ iff $(q,t,\beta)|=\phi$.

Proof: We use induction on n. Note that for any controller strategy (q,t) and any (q,t)-allowable strings α, β, the set of formulas ϕ such that $(q,t,\alpha)|=\phi$ iff $(q,t,\beta)|=\phi$ is closed under Boolean combinations. Hence, the only rank n formulas we really need to look at are the basic ones -- the elements of $\mathbf{B}_n$. Moreover, by symmetry, it is enough to show that if $\alpha\sim_{n,(q,t)}\beta$ and $(q,t,\alpha)|=\phi$, where $\phi\in\mathbf{B}_n$, then $(q,t,\beta)|=\phi$.

We start with n=0. Suppose $\alpha\sim_{0,(q,t)}\beta$ and $(q,t,\alpha)|=p$, where $p\in\mathbf{A}\cup\mathbf{Q}$. If $p\in\mathbf{A}$, then by the definition of satisfaction, $p=\alpha_0$. By $\sim_{0,(q,t)}$-equivalence, $\alpha_0=\beta_0$, so $(q,t,\beta)|=p$. If $p\in\mathbf{Q}$, then by the definition of satisfaction, p=q, and then $(q,t,\beta)|=p$. Therefore the lemma holds for n=0. Supposing that the lemma holds for n, consider n + 1. Let ϕ have rank n and suppose that $\alpha\sim_{n+1,(q,t)}\beta$. If $(q,t,\alpha)|=O\phi$, then by the definition of satisfaction, $(q^{\alpha,t,1},t,\alpha^1)|=\phi$. By $\sim_{n+1,(q,t)}$-equivalence, $q^{\alpha,t,1}=q^{\beta,t,1}$ and $\alpha^1\sim_{n,(s,t)}\beta^1$, where $s=q^{\alpha,t,1}$. Then by the Induction Hypothesis, we have $(q^{\beta,t,1},t,\beta^1)|=\phi$, so $(q,t,\beta)|=O\phi$. Let $(q,t,\alpha)|=\Box\phi$. By the definition of satisfaction, for all $i\geq 0$, $(q^{\alpha,t,i},t,\alpha^i)|=\phi$. By $\sim_{n+1,(q,t)}$-equivalence, for each $j\geq 0$, there is some $i\geq 0$ such that $q^{\alpha,t,i}=q^{\beta,t,j}$ and $\alpha^i\sim_{n,(s,t)}\beta^j$, where $s=q^{\alpha,t,i}$. Then by the Induction Hypothesis, $(q^{\beta,t,j},t,\beta^j)|=\phi$, so $(q,t,\beta)|=\Box\phi$. Finally, let $(q,t,\alpha)|=\Diamond\phi$. There is some $i\geq 0$ such that $(q^{\alpha,t,i},t,\alpha^i)|=\phi$. Then for some $j\geq 0$, $q^{\alpha,t,i}=q^{\beta,t,j}$ and $\alpha^i\sim_{n,(s,t)}\beta^j$, where $s=q^{\alpha,t,i}$, so by the Induction Hypothesis, $(q^{\beta,t,j},t,\beta^j)|=\phi$. Therefore, $(q,t,\alpha)|=\Diamond\phi$. This completes the proof.

A string $\alpha\in\mathbf{A}^\omega$ is said to be ultimately periodic, with period p and onset of periodicity u, if α has the form υ^τ^τ^τ^ ..., where υ and τ are finite strings, of lengths u and p, respectively. We say that α is determined by the pair (υ,τ). For ultimately periodic strings, it is relatively easy to answer questions about satisfaction.

Lemma 2. We can effectively decide, given a formula ϕ, a controller strategy (q,t), and a pair of finite strings (υ,τ) determining a string $\alpha\in\mathbf{A}^\omega$, whether $(q,t,\alpha)|=\phi$.

Proof: First of all, note that we can tell whether the ultimately periodic string α determined by (υ,τ) is (q,t)-allowable. Now, we proceed by induction on formulas ϕ, describing a method for deciding whether $(q,t,\alpha)|=\phi$ if α is a (q,t)-allowable string determined by (υ,τ). We start with ϕ of the form p for $p\in\mathbf{A}\cup\mathbf{Q}$. Let α be determined by (υ,τ). If $p\in\mathbf{A}$, then $(q,t,\alpha)|=p$ iff $p=\alpha_0$, where α_0 is the first symbol of υ^τ. If $p\in\mathbf{Q}$, then $(q,t,\alpha)|=p$ iff p=q. There is no difficulty in deciding satisfaction for Boolean combinations of formulas that we can deal with, so let us turn to the temporal operators.

Suppose that we have a procedure for deciding satisfaction for ϕ. We must say how to deal with $O\phi$, $\Box\phi$, and $\Diamond\phi$. We consider $O\phi$ first. If α is the ultimately periodic string determined by (υ,τ), then α^1 is also ultimately periodic, determined by (υ^1,τ), where υ^1 is the result of dropping the first term of υ if $\upsilon\neq\emptyset$, and of τ if $\upsilon=\emptyset$. We have $q^{\alpha,t,1}=t(q,\alpha_0)$. By the definition of satisfaction, $(q,t,\alpha)|=O\phi$ iff $(q^{\alpha,t,1},t,\alpha^1)|=\phi$, and we know how to decide this. Next, consider $\Box\phi$. Let υ^τ have length m. By the definition of satisfaction, $(q,t,\alpha)|=\Box\phi$ iff for all $i\geq 0$, $(q^{\alpha,t,i},t,\alpha^i)|=\phi$. For each $i\geq 0$, there is some $j<m$ such that $q^{\alpha,t,i}=q^{\alpha,t,j}$ and $\alpha^i=\alpha^j$, so it is enough to check that $(q^{\alpha,t,j},t,\alpha^j)|=\phi$ for all $j<m$. In terms of (q,t) and (υ,τ), we can figure out what $q^{\alpha,t,j}$ is, and we can also find a variant υ^j of υ such that (υ^j,τ) determines α^j. By our Induction Hypthesis, we can decide whether ϕ is satisfied by the triples $(q^{\alpha,t,j},t,\alpha^j)$ for all j<m. Finally, consider $\Diamond\phi$. If υ^τ has length m, then we have $(q,t,\alpha)|=\Diamond\phi$ iff $(q^{\alpha,t,j},t,\alpha^j)|=\phi$ for some j<m, and we can check this. This completes the proof of the lemma.

If (q,t) is a controller strategy and ϕ is a formula, then (q,t) is said to make ϕ valid if $(q,t,\alpha)|=\phi$ for all (q,t)-allowable strings α. There may in general be continuum many (q,t)-allowable strings, and only countably many are ultimately periodic. The next lemma says that to decide whether (q,t) makes ϕ valid, we do not need to look at all (q,t)-allowable α, or even all ultimately periodic α. It is enough to consider a particular finite collection with bounded period and onset of periodicity. The lemma may be of some interest apart from its use in proving decidability.

Lemma 3. For any controller strategy (q,t), any $n \in \omega$, and any (q,t)-allowable string α, there is an ultimately periodic string β such that $\alpha \sim_{n,(q,t)} \beta$. Moreover, we can compute bounds on the onset and period in terms of n.

Proof: Let $|X|$ denote the number of elements in set X . We shall define L_n and K_n for $n \in \omega$ such that K_n is an upper bound on the number of $\sim_{n,(q,t)}$-classes (for any controller strategy (q,t)), and for any (q,t)-allowable string α, there exist υ and τ such that $\upsilon{}^\wedge\tau$ has length at most L_n and (υ,τ) determines a (q,t)-allowable string β such that $\beta \sim_{n,(q,t)} \alpha$. For n=0, we can take K_0 to be $|A|$ (since the $\sim_{0,(q,t)}$-class of α is determined by the first symbol of α). We must say what L_0 is. For any (q,t)-allowable string α, there exist $i<j\leq|Q|\cdot|A|$ such that $q^{\alpha,t,i} = q^{\alpha,t,j}$ and $\alpha_i=\alpha_j$. Let $\alpha \restriction k$ denote the finite string consisting of the first k symbols of α. If $\upsilon=\alpha\restriction i$ and $\upsilon{}^\wedge\tau=\alpha\restriction j$, then (υ,τ) determines an ultimately periodic (q,t)-allowable β such that $\beta\sim_{0,(q,t)}\alpha$. Therefore, we can take L_0 to be $|Q|\cdot|A|$.

Suppose that for $k \leq n$ the lemma holds and we have determined K_k and L_k. If α is (q,t)-allowable, and $q'=q^{\alpha,t,i}$, we write $C_k^{\alpha,i}$ for the $\sim_{k,(q',t)}$-class of α^i. We may refer to the pair $(q^{\alpha,t,i},C_k^{\alpha,i})$ as the k-class of α_i. Note that for two (q,t)-allowable strings α and β, $\alpha \sim_{n+1,(q,t)} \beta$ iff the following three conditions hold:

(1) $C_n^{\alpha,0} = C_n^{\beta,0}$,

(2) $C_n^{\alpha,1} = C_n^{\beta,1}$, and

(3) $\{ (q^{\alpha,t,i},C_n^{\alpha,i}) : i\in \omega \} = \{ (q^{\beta,t,i},C_n^{\beta,i}) : i\in\omega \}$.

Condition (1) says that $\alpha \sim_{n,(q,t)} \beta$, and this implies that $q^{\alpha,t,1} = q^{\beta,t,1}$. If $q' = q^{\alpha,t,1} = q^{\beta,t,1}$, then Condition (2) says that $\alpha^1 \sim_{n,(q',t)} \beta^1$. Condition (3) says that for each i there is some j (and also for each j there is some i) such that $q^{\alpha,t,i} = q^{\beta,t,j}$ and if q' is this common state, then $\alpha^i\sim_{n,(q',t)} \beta^j$. This analysis shows that we can take K_{n+1} to be $K_n^2\cdot 2^r$, where $r=K_n\cdot|Q| - 1$.

Next, we describe a method for finding an ultimately periodic string β that is $\sim_{n+1,(q,t)}$-equivalent to a given string α, with respect to a given controller strategy (q,t). Having done this, we shall be able to say what L_{n+1} is. There are at most $|Q|\cdot K_n$ different n-classes. We mark any α_i such that there is no $j < i$ with α_i and α_j representing the same n-class. It may be that some n-classes only occur finitely many times, but at least one occurs infinitely often in α. Let α_r be the first symbol representing an n-class that occurs infinitely often. Of course, α_r is marked. Let α_s be the first symbol after all of the marked ones such that α_s represents the same n-class as α_r.

If we let $\upsilon=\alpha\restriction r$, $\upsilon{}^\wedge\tau=\alpha\restriction s$, and γ is the ultimately periodic string determined by (υ,τ), then we could show (with effort) that $\gamma\sim_{n+1,(q,t)}\beta$. However, $\upsilon{}^\wedge\tau$ may be very long. We now reduce the length. Suppose that we have α_i and α_j representing the same n-class, where $i < j < s$, and for all k such that $i \leq k \leq j$, α_k is not marked. Then we close up the interval, leaving out of $\upsilon{}^\wedge\tau$ all α_k for $i < k \leq j$. When we have closed up as much as possible in this way, $\upsilon{}^\wedge\tau$ will consist of at most $|Q|\cdot K_n$ marked symbols plus at most $|Q|\cdot K_n$ additional symbols after each marked one. We take L_{n+1} to be $|Q|\cdot K_n + (|Q|\cdot K_n)^2$.

Let β be the ultimately periodic string determined by (υ,τ) (after the shortening process). All that remains to be done is to prove that $\alpha \sim_{n+1,(q,t)}\beta$. Each symbol β_m of β corresponds to a particular symbol α_i from α (one that was included in υ or τ).

Claim: If β_m corresponds to α_i, then (a) $q^{\alpha,t,i} = q^{\beta,t,m}$, and (b) for each $k \leq n$, $C_k^{\alpha,i} = C_k^{\beta,m}$.

Suppose for the moment that the claim is true. Then we can show easily that $\beta \sim_{n+1,(q,t)} \alpha$. Recall the three conditions above. Condition (1) will hold because β_0 corresponds to α_0. If we only marked one symbol, then α is constant (it consists of infinitely many repetitions of a single symbol) and $\beta=\alpha$. If there is more than one symbol marked, then we saved α_1 (whether or not we marked it) and β_1 corresponds to α_1. Therefore, Condition (2) holds. Condition (3) holds because we marked representatives of all n-classes in α. Therefore, if the claim is true, then $\beta \sim_{n+1,(q,t)} \alpha$, and to complete the proof of the lemma, all that remains is to prove the claim.

Proof of Claim: If β_m corresponds to α_i and β_{m+1} corresponds to α_j, there are three cases: (i) $j=i+1$, (ii) $j>i+1$, and (iii) $j<i$. Case (ii) indicates that we closed up the interval between α_i and α_j. Then α_i and α_{j-1} represent the same n-1-class. Case (iii) indicates that α_i is the last symbol of τ and α_j is the first (i.e., $i=s$ and $j=r$). Then α_{i+1} and α_j represent the same n-class.

It is easy to check that if β_m corresponds to α_i, then $q^{\alpha,t,i} = q^{\beta,t,m}$, and the symbols β_m and α_i match. Therefore, we have part (a) of the claim and part (b) for k=0. We continue with the proof of (b) by induction on k. We show that if (b) holds for k, where $k < n$, then it also holds for $k + 1$. Let β_m correspond to α_i. To show that β_m and α_i represent the same k+1-class, we verify the three conditions given earlier. By the Induction Hypothesis, β_m and α_i represent the same k-class. (This is Condition (1)).

To verify Condition (2), we consider β_{m+1} and show that in the three cases (i), (ii), and (iii) above, β_{m+1} and α_{i+1} represent the same k-class. Suppose β_{m+1} corresponds to α_j. By the Induction Hypothesis, β_{m+1} and α_j represent the same k-class, so we will be done if we can show that α_j and α_{i+1} represent the same k-class. This is trivially true if $j=i+1$. If $j > i+1$, then we are in case (ii) described above, and α_{i+1} and α_j represent the same n-1-class. Since $k \leq n - 1$, it follows that α_{i+1} and α_j represent the same k-class. Finally, if $j < i$, then we are in case (iii), and again α_{i+1} and α_j represent the same n-class. We have shown that in all cases β_{m+1} and α_{i+1} represent the same k-class.

To prove Condition (3), we must show that the k-classes represented by $\beta_{m'}$ for $m' \geq m$ are the same as the k-classes represented by $\alpha_{i'}$ for $i' \geq i$. First, take $i' \geq i$. If τ includes some α_j representing the same k-class as α_i, then we have $\beta_{m'}$ representing this k-class for arbitrarily large m'. Suppose that τ does not include any such α_j. It will follow that the k-class of $\alpha_{i'}$ is represented only finitely many times in α. The reason for this is as follows: If the k-class were represented infinitely often, then some n-class in the k-class would be represented infinitely often. Let $\alpha_{i''}$ be the first representative of this k-class. By our choice of r, $r \leq i''$, and we would have the k-class represented in τ, a contradiction. Since the k-class of $\alpha_{i'}$ is represented only finitely many times in α, there is a largest j such that α_j represents the k-class. If $j \geq r$, then for each of the infinitely many $r' > r$ such that $\alpha_{r'}$ represents the same n-class as α_r, there would be $j' \geq r'$ representing the same n-1-class, and, hence, the same k-class, as α_j. This cannot happen, so we have $j < r$.

Since α_j is the last representative of its k-class, it is also the last representative of its n-1-class. Then it is easy to check that α_{j+1} is the first representative of its n-class. This means that α_{j+1} is marked and corresponds to some (first) element of β, which we call $\beta_{m'+1}$. (The only way there could be more than one element of β corresponding to α_{j+1} is if $j+1=r$.) The symbol $\beta_{m'+1}$ has an immediate predecessor, which we call $\beta_{m'}$. We have $i \leq i' \leq j < j+1 \leq r$, and $\beta_{m'}$ corresponds to some $\alpha_{j'}$, where $i \leq j' \leq j$. (See Figure 3.1 below.)

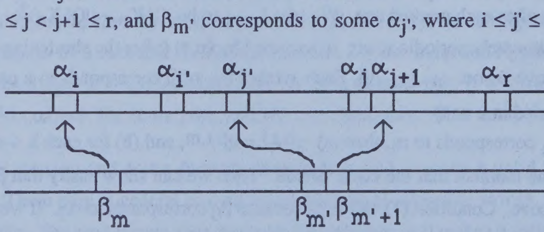


Figure 3.1

We may be lucky and have j'=j. If j' < j, then we closed up the interval between $\alpha_{j'}$ and α_{j+1}. Then $\alpha_{j'}$ and α_j represent the same n-class and, hence, the same k-class.

Now, take m' ≥ m. We must find i' ≥ i such that $\beta_{m'}$ and $\alpha_{i'}$ represent the same k-class. Let $\beta_{m'}$ correspond to α_j. Then $\beta_{m'}$ and α_j represent the same k-class. If j ≥ i, then we are done. Suppose j < i. This means that β_m and $\beta_{m'}$ lie in different copies of τ, and $r \leq j < i < s$. Since α_r and α_s represent the same n-class, there must be i' ≥ s such that α_j and $\alpha_{i'}$ represent the same k-class (the same n - 1-class, in fact). We have now verified Condition (3), completing the proof that if β_m corresponds to α_i, then the k+1-classes match. This was all that remained to prove the lemma.

Theorem 4 (Decidability). We can effectively decide whether a controller strategy (q,t) makes ϕ valid.

Proof: The decision procedure is as follows: First, list the pairs (υ,τ) determining the ultimately periodic (q,t)-allowable strings α with onset and period bounded in terms of n, as in Lemma 3. Then test whether $(q,t,\alpha) \models \phi$ for these α, as in Lemma 2. By Lemmas 1 and 3, this is enough to decide whether (q,t) makes ϕ valid.

Theorem 5 (Synthesis). Given a formula ϕ, we can effectively either find a controller strategy (q,t) making ϕ valid, or else say for sure that no such strategy exists.

Proof: Since there are only finitely many controller strategies (q,t), this follows from Theorem 4.

The theorems above say that we can, in principle, do a really thorough job of control system analysis and also mechanically carry out controller synthesis for the type of plant that we have described.

§ 4. Control System Analysis and Synthesis Examples

In this section we describe in detail two examples and say for these examples what the results of the last section mean. The first example, from chemical process control, is the surge tank described in the Introduction. The plant has controller states as inputs and sensor symbols as outputs. Let the set of possible initial symbols be I={a_1,a_3} (liquid level empty or normal). Figure 4.1 gives the relation N. The body of the table gives the possible next symbols that could result from a given current sensor symbol and controller state (i.e. state assumed in response to that symbol). For example, if the current sensor symbol is a_3 (normal liquid level) and the next controller state is q_1 (fill valve open), then by the next second, the output may be a_3 still, since some unpredictable users may be drawing liquid from the tank, or the ouput may be a_4 (high), if the present level is approaching high.

		Plant				
N		Sensor Symbol				
		a_1	a_2	a_3	a_4	a_5
Controller State	q_0	a_1	a_1,a_2	a_2,a_3	a_3,a_4	a_4,a_5
	q_1	a_2	a_2,a_3	a_3,a_4	a_4,a_5	a_5
		Possible next symbols				

Figure 4.1

The controller has the sensor symbols as inputs and states as outputs. Figure 4.2 gives the transition function t, taking the current state and sensor symbol to the next state. For example, if the current state is q_0 (fill valve closed) and the sensor symbol is a_2 (low), then the next input to the plant is q_1 (fill valve open). Let the initial state be q_0.

		Controller				
t		Sensor Symbol				
		a_1	a_2	a_3	a_4	a_5
Controller State	q_0	q_1	q_1	q_1	q_0	q_0
	q_1	q_1	q_1	q_1	q_0	q_0
		Next state				

Figure 4.2

Let's see what our results mean for rank 1 formulas. By Lemma 3, for any (q,t)-allowable string α, there is $\beta \sim_{1,(q,t)} \alpha$ such that β is ultimately periodic, and Lemma 1 says that for any rank 1 formula ϕ, $(q,t,\alpha) \models \phi$ iff $(q,t,\beta) \models \phi$. We have in mind here that $q=q_0$. Notice that the string $a_1a_2a_3a_4a_3a_3a_3a_4a_3a_3a_3a_3a_4a_3\ldots$ is not ultimately periodic since the symbol a_3 repeats once, twice, three times, etc. However, it is $\sim_{1,(q,t)}$-equivalent to the string $a_1a_2a_3a_4a_3a_4a_3a_4\ldots$. We identify the pair (υ,τ) with the string $\beta=\upsilon\hat{}\tau\hat{}\tau\hat{}\tau\ldots$ determined by it. The proof of Lemma 3 tells us that we can obtain representatives of all $\sim_{1,(q,t)}$-equivalence classes by looking at all (υ,τ) pairs such that $\upsilon\hat{}\tau$ has length $\leq L_1=110$ and dropping those which do not determine a (q,t)-allowable string. Here is a simpler list, with just one representative of each $\sim_{1,(q,t)}$-class: $(\emptyset,a_3)$, (a_3,a_3a_4), (a_3a_3,a_4), (a_3a_4,a_3), (a_3a_4,a_4), $(a_3a_4a_3a_3,a_4)$, $(\emptyset,a_3a_4)$, (a_1,a_2), (a_1a_2,a_3), (a_1a_2,a_3a_4), $(a_1a_2a_3a_4,a_4a_3)$, $(a_1a_2a_3,a_4)$.

One particular closed loop specification that we would like to meet is the rank 1 formula, $\phi=a_3\rightarrow\Box(\neg(a_1\vee a_5))$, which says "if the liquid level starts off normal, then it will always be the case that the liquid level will not be empty or full". To check whether the controller strategy (q,t) makes our specification $\phi=a_3\rightarrow\Box(\neg(a_1\vee a_5))$ valid, we check that $(q,t,\alpha)\models\phi$ for those α determined by the (υ,τ) pairs on our list. The length of $\upsilon\hat{}\tau$ is some $m\leq 6$. By the definition of satisfaction, it is enough to show $(q^{\alpha,t,i},t,\alpha^i)\models\neg(a_1\vee a_5)$ for all $i\geq 0$. From the proof of Lemma 2, this is equivalent to showing that $(q^{\alpha,t,i},t,\alpha^i)\models\neg(a_1\vee a_5)$ for all $i\leq m$. By the definition of satisfaction, it is enough to show that it is not the case that $(q^{\alpha,t,i},t,\alpha^i)\models a_1$ or $(q^{\alpha,t,i},t,\alpha^i)\models a_5$ for all $i\leq m$. By examining the symbols in each of the (υ,τ) pairs above, we see that the formula is valid.

Another specification is the rank 2 formula, $a_1\rightarrow\Diamond\Box(a_3\vee a_4)$, which says "if initially the liquid level is empty, then eventually we will always have a liquid level of normal or high". We now have additional (υ,τ) pairs to check since certain pairs which were $\sim_{1,(q,t)}$-equivalent to the ones above, are not $\sim_{2,(q,t)}$-equivalent. For example, (a_3a_3,a_4) is $\sim_{1,(q,t)}$-equivalent to $(a_3a_3a_3,a_4)$ but not $\sim_{2,(q,t)}$-equivalent. Similarly, $(a_3a_3,a_4a_3a_3)$ and $(a_3a_3a_3,a_4a_3a_3)$ are both $\sim_{1,(q,t)}$-equivalent to (a_3,a_3a_4) but not $\sim_{2,(q,t)}$-equivalent to (a_3,a_3a_4) or to each other. Further $\sim_{2,(q,t)}$-equivalent classes are represented by $(a_3a_3,a_4a_3a_3)$, $(a_3a_4a_3,a_4)$, $(a_3a_3a_4,a_4a_3a_3a_4)$, $(a_1a_2a_2,a_3)$,and $(a_1a_2,a_3a_4a_4a_4a_3)$. We are not giving a complete list here, but there is a mechanical procedure for doing so. With higher rank closed loop specification formulas, we need to examine more precise plant input/output behavior.

The specification $\phi=a_1\rightarrow\Diamond\Box(a_3\vee a_4)$ is met iff $(q,t,\alpha)\models\phi$ for all α determined by the (υ,τ) pairs on our extended list. We follow Lemma 2. By the definition of satisfaction, to test a given α (determined by some (υ,τ) on our list), it is enough to show that if the first symbol of α is a_1, then $(q,t,\alpha)\models\Diamond\Box(a_3\vee a_4)$. By the proof of Lemma 2 and the definition of satisfaction, it is enough to show that τ has no other symbols than a_3 and a_4. When we come to the pair (a_1,a_2), we find that the formula is invalid. The string $a_1a_2a_2a_2a_2\ldots$ determined by this pair represents a type of behavior that the plant might exhibit so that the closed loop specification formula is not met; i.e., it is a counterexample.

Next, we turn to controller synthesis. Suppose we wish to satisfy $\phi=a_1\rightarrow\Diamond\Box(a_2\vee a_3)$, i.e. "if initially the tank is empty, then eventually there will be a time such that from that time on, the liquid level will be low or normal". This rank 2 formula is invalid for the controller strategy (q,t) above, with a counterexample determined by the pair $(a_1a_2a_3a_4,a_4)$. Theorem 5 says that an exhaustive search will produce a controller strategy to make ϕ valid if one exists. As a synthesis heuristic, we examine some of the (υ,τ) pairs used in the above analysis. For the string $(a_1a_2a_3a_4,a_4)$, the sequence of associated states is $q_0q_1q_1q_1q_0q_0q_0q_0\ldots$. With this and the relation N, we see that if we changed t so that the controller responded to the output symbol a_3 with a q_0 instead of a q_1, the specification might be met. Name this new transition function t'. For this t' and $q=q_0$ as above, some of the (υ,τ) pairs representing different $\sim_{2,(q,t')}$-equivalence classes are: (a_1,a_2), $(a_1,a_2a_3a_3a_2)$, $(a_1,a_2a_2a_3a_2)$, $(a_1a_2a_3,a_3)$, $(a_1a_2a_2,a_3)$, $(a_1a_2,a_3a_3a_2)$, $(a_1a_2,a_3a_3a_3a_2)$, $(a_1a_2,a_3a_3a_2a_2)$, $(a_1a_2,a_3a_3a_3a_2a_2)$. Examining all the (υ,τ) pairs would show that the synthesized controller strategy (q,t') makes the formula valid. (We could shorten the procedure.)

We use a manufacturing system for the second example. We have one machine which must process different parts from two buffers (B1 and B2) each supplied by two different producers (P1 and P2). Upon completion of processing, each type of part is put in its respective output bin. A controller must ensure mutual exclusion in the machine, i.e., only one part of one type is processed at a time. The controller must also ensure that producer P1 gets priority in the use of the machine. To perform its task, the controller can sense whether there is a part in a given buffer and can also determine the type of part in the machine. The controller acts to allow the different parts to enter the machine.

We use a 4 bit binary string to represent each of the sixteen different sensor symbols. Let the first bit be 1 if there is a P1 part in the machine and 0 otherwise. The second bit indicates the presence or absense of a P2 part in the machine. The third and fourth bits represent the presence or absence of parts in buffers B1 and B2, respectively. For example, the symbol "1001" indicates that 1) there is a P1 part in the machine, 2) there isn't a P2 part in the machine, 3) there isn't a P1 part in buffer B1, and 4) there is a P2 part in buffer B2. Let the states of the controller be q_1 (allowing a part from P1, presently in B1, to enter the machine), q_2 (allowing a part from P2, presently in B2, to enter the machine), and q_3 (not allowing any parts to enter the machine). The controller's schedule is such that the controller is alerted when the sensor symbol changes and responds then. In this case, O means "next time something happens". We have, $A=\{0000,0001,0010,0011,0100,0101,0110, 0111,1000,1001,1010,1011,1100,1101,1110,1111\}$ and $Q=\{q_1,q_2,q_3\}$.

To avoid confusion, we change our notation for strings, writing a_1,a_2,a_3 instead of $a_1a_2a_3$ since each a_i is a binary string of length 4. We also write $(\upsilon;\tau)$ instead of (υ,τ). For this plant, the relation N and the transition function t are shown in Figures 4.3 and 4.4, respectively. We have $I=\{0000\}$ and the initial state is $q=q_3$. There are two closed loop specification formulas. The mutual exclusion specification, which we verify first, is $\phi=\Box(\neg(1100\vee1101\vee1110\vee1111))$. Some typical (υ,τ) pairs which must be examined are: $(\emptyset;0000,0001,0100)$, $(0000;0001,0101)$, $(0000,0010;1000)$, $(0000,0011;1011)$. (These represent distinct $\sim_{1,(q,t)}$-equivalence classes.) Let α be one of the (υ,τ) pairs, m be the length of $\upsilon\hat{}\tau$. To show that $(q,t,\alpha)\models\phi$ (following Lemma 2, and the definition of satisfaction), it is enough to test that for all $j\leq m$ $(q^{\alpha,t,j},t,\alpha^j)\models1100$ or $(q^{\alpha,t,j},t,\alpha^j)\models1101$ or $(q^{\alpha,t,j},t,\alpha^j)\models1110$ or $(q^{\alpha,t,j},t,\alpha^j)\models1111$. For the (υ,τ) pairs above and all others this is satisfied. Hence, the formula is valid.

Plant				
N		Controller State		
		q_1	q_2	q_3
Sensor Symbol	0000	-	-	0000,0001,0010,0011
	0001	-	0100,0101,0110,0111	0001,0011
	0010	1000,1001,1010,1011	-	0010,0011
	0011	1001,1011	0110,0111	0011
	0100	-	-	0100,0000
	0101	-	0100,0101,0110,0111	0001,0011,0101,0111
	0110	1000,1001,1010,1011 1100,1101,1110,1111	-	0010,0011,0110,0111
	0111	1001,1011,1101,1111	0010,0011,0110,0111	0111,0011
	1000	-	-	0000,0001,0010,0011, 1000,1001,1010,1011
	1001	-	0100,0101,0110,0111 1100,1101,1110,1111	0001,0011,1011,1001
	1010	1000,1001,1010,1011	-	0010,0011,1010,1011
	1011	1001,1011,1101,1111	0100,0101,0110,0111 1100,1101,1110,1111	1011,0011
	1100	-	-	(all symbols possible)
	1101	-	0100,0101,0110,0111 1100,1101,1110,1111	0001,0011,0101,0111 1001,1011,1101,1111
	1110	1000,1001,1010,1011 1100,1101,1110,1111	-	0010,0011,0110,0111 1010,1011,1110,1111
	1111	1001,1011,1101,1111	0110,0111,1110,1111	0011,0111,1011,1111
		Possible next symbols		

Figure 4.3

Controller																	
t		Sensor Symbol															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Controller State	q_1	q_3	q_2	q_1	q_1	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3
	q_2	q_3	q_2	q_1	q_1	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3
	q_3	q_3	q_2	q_1	q_1	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3	q_3
		Next state															

Figure 4.4

The priority specification is $\phi=\Box(0011\rightarrow O(1001\vee 1011))$. Since this has rank 2, the mechanical procedure would say to examine (υ,τ) pairs representing all $\sim_{2,(q,t)}$-equivalence classes. In addition to the list above, we must also examine the (υ,τ) pairs: (0000,0000,0011;1011,1011,0011), (0000;1011,1011,0011), (0000,0000;1001,0011), and many others. We could use the mechanical procedure to test the formula ϕ, but it is clear just from examining N and t that the specification is met.

As in the surge tank example, we could take new closed loop specifications and synthesize controller strategies to meet them, if possible. For example, we might change the priority from producer P1 to P2. The manufacturing example above is similar to the "Two Class Parts Processing" example in [3]. There, however, Thistle and Wonham allow an arbitrary finite number of parts of one type to enter the machine. This forces their controller to have an infinite number of states, so that our decision procedure does not apply to their example. There are, however, many practical problems that are finite state.

REFERENCES

[1] Manna Z.,Wolper P., "Synthesis of Communicating Processes from Temporal Logic Specifications", Dept. of Computer Science, Stanford Univ., Report No. STAN-CS-81-872.

[2] Buchi J.R., "On a Decision Method in Restricted Second Order Arithmetic",Proc. of 1960 Int. Congress of Logic, Methodology, and Philosophy of Science, ed. by Nagel, Suppes, & Tarski, Stanford Univ. Press, 1962, pp. 1-11.

[3] Thistle J. G., Wonham W. M., "Control Problems in a Temporal Logic Framework", International Journal Control, 1986, Vol 44, No. 4 pp 943 - 976.

[4] Ostroff J. S., Real-Time Computer Control of Discrete Systems Modelled by Extended State Machines: A Temporal Logic Approach, PhD Dissertation, Report No. 8618, Dept. of Elect. Eng., Univ. of Toronto, Jan. 1987.

A LINEAR TIME ALGORITHM FOR CIRCULAR PERMUTATION LAYOUT[1]

CHONG S. RIM, NICHOLAS J. NACLERIO[2]
Electrical Engineering Department and Systems Research Center
University of Maryland, College Park, MD 20742
SUMIO MASUDA[3]
Department of Information and Computer Sciences
Osaka University, Toyonaka, Osaka 560, Japan
KAZUO NAKAJIMA
Electrical Engineering Department, Institute for Advanced Computer Studies,
and Systems Research Center
University of Maryland, College Park, MD 20742

ABSTRACT

Suppose that two sets of terminals $t_1, t_2, \ldots, t_n$ and $b_1, b_2, \ldots, b_n$ are located on two concentric circles C_{out} and C_{in}, respectively. Given a permutation π of integers $1, 2, \ldots, n$, the circular permutation layout problem is a single layer routing problem which is to connect each pair of terminals t_i and $b_{\pi(i)}$ for $i = 1, 2, \ldots n$. In this paper, we present a linear time algorithm for the following case: (i) no wire can cross C_{out}, (ii) at most one wire can pass between any two adjacent terminals on C_{in}, and (iii) no wire can cross C_{in} more than once. The best previously known algorithm for the same case has time complexity $O(n^2)$.

1. INTRODUCTION

Suppose that two sets of terminals $t_1, t_2, \ldots t_n$ and $b_1, b_2, \ldots, b_n$ are located on two concentric circles C_{out} and C_{in}, respectively. We assume that the circle C_{out} is outside the circle C_{in} and that the terminals are labeled in ascending order of their subscripts in the clockwise direction. Given a permutation π of integers $1, 2, \ldots, n$, the *circular permutation layout* (CPL) problem is the problem of connecting each pair of terminals t_i and $b_{\pi(i)}$ for $i = 1, 2, \ldots, n$ with zero width wires such that no two wires which correspond to different terminal pairs intersect each other.

The CPL problem was first proposed by Osawa [6] as an extension of the *linear permutation layout* (LPL) problem. In the LPL problem, two sets of terminals $t_1, t_2, \ldots, t_n$ and $b_1, b_2, \ldots, b_n$ are located, from left to right, on two parallel lines L_t and L_b, respectively. This problem has widely been studied [2,3,4,5,8,9].

The CPL problem may arise in the design of printed circuit boards (PCBs) and hybrid integrated circuits. Consider for example, the problem of connecting a series of incoming wires to the pins on a particular module on a PCB (see Fig. 1). We assume that no two pins on the module are electrically equivalent and hence any two wires for the connections must not intersect each other. The order of the incoming wires may not be the same as that of the pins on the module because of previous routing of the incoming wires. If only one layer is available to realize the connections, this problem is equivalent to the CPL problem. The terminals on C_{in} correspond to the pins on the module and the terminals on C_{out} represent the incoming wires. Since the pins have fixed spacing and the wires have finite width, we assume that the number of wires which can pass between two adjacent pins is limited to one. No wires may cross the boundary represented by C_{out} since they would intersect the existing incoming wires.

From the observations mentioned above, we consider the CPL problem with the following constraints on the wires in its solution layout [6]:

1. No wire can cross C_{out},
2. At most one wire can pass between any two adjacent terminals on C_{in}, and
3. No wire can cross C_{in} more than once.

We call a layout in which wires satisfy the above constraints a *CPL1-solution*. For example, Fig. 2 shows a CPL1-solution for the permutation π = 10 7 6 4 3 5 2 9 8 1 16 12 15 13 14 11. For simplicity,

---

[1] This work was supported in part by National Science Foundation grants MIP-84-51510 and CDR-85-00180 and a grant from AT&T.

[2] N. J. Naclerio was also supported during the course of this work by a fellowship from Westinghouse Electric Corporation.

[3] The work of S. Masuda was done while he was visiting the Electrical Engineering Department and Systems Research Center at the University of Maryland, College Park, MD 20742.

we label each terminal t_i or b_i as i in the figures throughout this paper. Ozawa [6] developed an $O(n^2)$ time algorithm for finding a CPL1-solution, where n is the number of terminal pairs to be connected.

In this paper, we present an $O(n)$ time algorithm for the CPL problem under the constraints listed above. Under similar constraints, the LPL problem can also be solved in $O(n)$ time using our algorithm.

In Section 2 we introduce some basic definitions and notation. Section 3 describes some useful properties of a CPL1-solution. In Section 4 we explain merging operations, which play an important role in our algorithm. The algorithm is presented in Section 5. Because of the space limitation, most proofs are omitted.

2. DEFINITIONS AND NOTATION

The problem which we consider in this paper is described as follows: "*Given a permutation π of integers $1, 2 \ldots, n$, find a CPL1-solution for it if one exists.*" If there exists a CPL1-solution for π, we say that π is *realizable.*

A *net* is an ordered pair of terminals $(t_i, b_{\pi(i)})$ which must be electrically connected. Let $N(\pi) = \{n_i = (t_i, b_{\pi(i)}) \mid 1 \leq i \leq n\}$ be the set of nets for a given permutation π.

Assume that π is realizable and let SL be any CPL1-solution for it. We denote by w_i the wire in SL which realizes net $n_i = (t_i, b_{\pi(i)})$. If w_i is located entirely within the region between the circles C_{in} and C_{out}, it is called a *direct wire*; otherwise it is called an *indirect wire*. For example, w_{15} is a direct wire and w_{16} is an indirect wire in Fig. 2. We assume that a net is routed by an indirect wire in SL if and only if it can not be routed with a direct wire.

We define $l \oplus 1$ to be $l+1$ if $1 \leq l \leq n-1$ and 1 if $l = n$. Because of the third constraint on a CPL1-solution, each indirect wire w_i in SL crosses C_{in} exactly once between some pair of terminals b_j and $b_{j\oplus 1}$, which we denote by $w_i \wr \bullet\, b_{j\oplus 1}$. For example, $w_1 \wr \bullet\, b_1$ in Fig. 2.

Let $p_1, p_2, \ldots, p_k$ be distinct integers between 1 and n. A sequence $[p_1, p_2, \ldots, p_k]$ is called an *increasing consecutive sequence* (**icseq**) if $p_i \oplus 1 = p_{i+1}$ for $i = 1, 2, \ldots, k-1$. An **icseq** is said to be *maximal* if and only if it is not contained in any other **icseq**.

Let $M = \{n_{p_i} = (t_{p_i}, b_{q_j}) | 1 \leq i, j \leq k, k \leq n\}$ be a nonempty subset of $N(\pi)$ such that $[p_1, p_2, \ldots, p_k]$ and $[q_1, q_2, \ldots, q_k]$ are icseqs. We call M a *cluster* in $N(\pi)$ if $\pi(p_i) = q_{k-i+1}$ for $1 \leq i \leq k$. If a cluster consists of only one net, it is called a *trivial cluster*; otherwise it is called a *nontrivial cluster*. A cluster is *maximal* if and only if it is not contained in any other cluster. For example, the instance shown in Fig. 2 has the following eleven maximal clusters: $\{n_2, n_3\}$, $\{n_4, n_5\}$, $\{n_6\}$, $\{n_7\}$, $\{n_8, n_9\}$, $\{n_{10}, n_{11}\}$, $\{n_{12}\}$, $\{n_{13}\}$, $\{n_{14}\}$, $\{n_{15}\}$ and $\{n_{16}, n_1\}$. It is easy to see that if two distinct clusters C and C' are both maximal, then $C \cap C' = \phi$. Furthermore, if $C \cup C' = N(\pi)$, then $N(\pi)$ itself is a cluster. Thus, if $N(\pi)$ is not a single cluster, it can uniquely be partitioned into three or more maximal clusters.

Let $M' = \{n_{p'_i} = (t_{p'_i}, b_{q'_j}) | 1 \leq i, j \leq m,\ m < n\}$ be another subset of $N(\pi)$ such that $[p'_1, p'_2, \ldots, p'_m]$ and $[q'_1, q'_2, \ldots, q'_m]$ are icseqs. We say that M is *parallel* to M', denoted by $M \parallel M'$, if and only if $p_k \oplus 1 = p'_1$ and $q_k \oplus 1 = q'_1$. For example, $\{n_4, n_5\} \parallel \{n_6\}$ and $\{n_{12}\} \parallel \{n_{13}, n_{14}, n_{15}\}$ in Fig. 2.

3. LAYOUT OF MAXIMAL CLUSTERS

Throughout this section, we assume that the given permutation π is realizable and that $N(\pi)$ has three or more maximal clusters. Let SL be any CPL1-solution for π.

Let $C = \{n_{p_i} = (t_{p_i}, b_{q_{k-i+1}}) \mid 1 \leq i \leq k,\ k < n\}$ be a nontrivial maximal cluster in $N(\pi)$, where $[p_1, p_2, \ldots, p_k]$ and $[q_1, q_2, \ldots, q_k]$ are icseqs. It is clear from the definition of a cluster that the layout of C in SL has at most one direct wire. If a net in C is routed by a direct wire in SL, then the layout of C must be one of the following two types of layouts [7], where $\lceil x \rceil$ denotes the smallest integer which is not less than x.

1. A Type DL layout of C is the layout such that $w_{p_{\lceil (k+1)/2 \rceil}}$ is a direct wire and $w_{p_i} \wr \bullet\, b_{q_i}$ for $i \neq \lceil (k+1)/2 \rceil$.
2. A Type DR layout of C is the layout such that $w_{p_{\lceil k/2 \rceil}}$ is a direct wire and $w_{p_i} \wr \bullet\, b_{q_i \oplus 1}$ for $i \neq \lceil k/2 \rceil$.

For example, the layout of $\{n_8, n_9\}$ is a Type DL layout and the layout of $\{n_4, n_5\}$ is a Type DR layout in Fig. 2. We call the layouts defined above Type D layouts for simplicity.

We may sometimes be able to route the nets in C using only indirect wires. In this case, there must be another disjoint maximal cluster $C' = \{n_{p'_i} = (t_{p'_i}, b_{q'_{k-i+1}}) \mid 1 \leq i \leq m,\ m < n\}$, where $[p'_1, p'_1, \ldots, p'_m]$ and $[q'_1, q'_2, \ldots, q'_m]$ are icseqs, such that the layout of $C \cup C'$ is one of the following three types of layouts [7]:

1. A Type IL layout of $C \cup C'$ is the layout such that $m = k$, $w_{p_i} \wr \bullet\, b_{q'_i}$ and $w_{p'_i} \wr \bullet\, b_{q_i}$ for $1 \leq i \leq k$.
2. A Type IR layout of $C \cup C'$ is the layout such that $m = k$, $w_{p_i} \wr \bullet\, b_{q'_i \oplus 1}$ and $w_{p'_i} \wr \bullet\, b_{q_i \oplus 1}$ for $1 \leq i \leq k$.

3. A Type IM layout of $C \cup C'$ is the layout such that either $m = k - 1$, $w_{p_k} \wr \bullet b_{q'_m \oplus 1}$, and $w_{p_i} \wr \bullet b_{q'_i}$ and $w_{p'_i} \wr \bullet b_{q_{i+1}}$ for $1 \leq i \leq k-1$ or $m = k+1$, $w_{p'_m} \wr \bullet b_{q_k \oplus 1}$, and $w_{p_i} \wr \bullet b_{q'_{i+1}}$ and $w_{p'_i} \wr \bullet b_{q_i}$ for $1 \leq i \leq k$.

We call all of the layouts defined above Type I layouts. In Fig. 2, the layout of $\{n_{10}, n_{11}\} \cup \{n_{16}, n_1\}$ is a Type IL layout and that of $\{n_2, n_3\} \cup \{n_7\}$ is a Type IM layout. There is no Type IR layout in the figure, but if we routed the nets in such a way that $w_{10} \wr \bullet b_{11}$, $w_{11} \wr \bullet b_{12}$, $w_{16} \wr \bullet b_1$ and $w_1 \wr \bullet b_2$, then the layout of $\{n_{10}, n_{11}\} \cup \{n_{16}, n_1\}$ would become a Type IR layout.

When the layout of $C \cup C'$ is a Type I layout in SL, we say that C' (resp., C) is the *mate* cluster of C (resp., C'). For example, in Fig. 2, $\{n_7\}$ is the mate cluster of $\{n_2, n_3\}$ with respect to their Type IM layout in SL.

We now consider the layout of a trivial maximal cluster $C = \{n_p = (t_p, b_q) \mid n_p \in N(\pi)\}$. If n_p is routed by a direct wire, then the layout satisfies the conditions of both Type DL and Type DR layouts and hence we regard the layout as a Type D layout.

Suppose that n_p is routed by an indirect wire. Then, there may exist a mate cluster C' such that $1 \leq |C'| \leq 2$ and the layout of $C \cup C'$ is a Type I layout. If $|C'| = 1$, the layout of $C \cup C'$ is either a Type IL or Type IR layout. Otherwise, the layout of $C \cup C'$ is a Type IM layout. If there is no mate cluster for C in SL, then we call the layout of C a Type IM *layout by itself.* For example, the layout of $\{n_{13}\}$ in Fig. 2 is a Type IM layout by itself.

From the above discussion, we arrive at the following theorem.

Theorem 1. *Let C be a maximal cluster. In any CPL1-solution, the layout of C is a* Type D *layout or there exists a maximal cluster C' such that $|C| - 1 \leq |C'| \leq |C| + 1$ and $C \cup C'$ forms a* Type I *layout, or C forms a* Type IM *layout by itself.* □

4. MERGING OPERATIONS

Assume that $N(\pi)$ consists of three or more maximal clusters. Let $M = \{n_{p_i} = (t_{p_i}, b_{q_j}) \mid 1 \leq i, j \leq k,\ k \leq n\}$ be a set of maximal clusters in $N(\pi)$ such that $[p_1, p_2, \ldots, p_k]$ and $[q_1, q_2, \ldots, q_k]$ are icseqs. Let $B_M = \{b_{q_1}, b_{q_2}, \ldots, b_{q_k}\}$.

We call M a *component* if it satisfies the property that if π is realizable, then its layout in any CPL1-solution satisfies the following restriction $(\star)$:

$(\star)$ if a wire passes between two adjacent terminals, say b_j and $b_{j\oplus 1}$, on C_{in}, then $\{b_j, b_{j\oplus 1}\} \cap B_M \neq \phi$.

For example, in Fig. 2, $\{n_4, n_5, n_6\}$ is a component while $\{n_4, n_5, n_6, n_7\}$ is not. Note that if a maximal cluster is a component, its layout in any CPL1-solution must be either a Type DL or Type DR layout if it is nontrivial, and a Type D layout if it is trivial. We say that the nets in a component are, or simply a component is, *routable* if the wires in its layout satisfy both the constraints on a CPL1-solution and the restriction $(\star)$.

Suppose that M is routable and let L_M be any such layout that satisfies the routability conditions on M. If there is a wire w in L_M such that either $w \wr \bullet b_{q_1}$ or $w \wr \bullet b_{q_k \oplus 1}$, then we call w a *boundary wire.* The only nets whose corresponding wires may be boundary wires in L_M are n_{p_1} and n_{p_k}, and if so, they are routed in such a way that $w_{p_1} \wr \bullet b_{q_1}$ and $w_{p_k} \wr \bullet b_{q_k \oplus 1}$, respectively. For example, if M is a maximal cluster and its layout is a Type DL (resp., Type DR) layout, then w_{p_1} (resp., w_{p_k}) is the boundary wire for M.

Since the CPL1-solution may not be unique for a realizable permutation π, if we first consider routing the nets in a component M to construct a CPL1-solution, then there may be many possible layouts of M whose wires satisfy the restriction $(\star)$. But, if we only consider the necessity of boundary wires in such layouts, we can classify each component into one of the following types:

1. *Type 0* if no boundary wire is necessary.
2. *Type x* if exactly one of the wires w_{p_1} and w_{p_k} must be a boundary wire but either one may be selected.
3. *Type l* if w_{p_1} must be a boundary wire while w_{p_k} need not be.
4. *Type r* if w_{p_k} must be a boundary wire while w_{p_1} need not be.
5. *Type lr* if both w_{p_1} and w_{p_k} must be boundary wires.

Fig. 3 illustrates typical examples of the above five types and their symbolic representations. Note that if a component M is a trivial (resp., nontrivial) maximal cluster, it is of *Type 0* (resp., *Type x*).

A *merging operation* is an operation which creates a new enlarged component by combining a component, called the *center of activity* (COA), with some other component or maximal clusters. In the following, we define two types of merging operations.

Let $M = \{n_{p_i} = (t_{p_i}, b_{q_j}) \mid 1 \le i, j \le k,\ k < n\}$ and $M' = \{n_{p'_i} = (t_{p'_i}, b_{q'_j}) \mid 1 \le i, j \le m,\ m < n\}$ be subsets of $N(\pi)$ such that $M \| M'$, and $[p_1, p_2, \ldots, p_k]$, $[q_1, q_2, \ldots, q_k]$, $[p'_1, p'_2, \ldots, p'_m]$ and $[q'_1, q'_2, \ldots, q'_m]$ are icseqs. Suppose that both M and M' are components and they are routable. If w_{p_k} and $w_{p'_1}$ must be the boundary wires in the layouts of M and M', respectively, these wires have to pass between the adjacent terminals b_{q_k} and $b_{q'_1}$. Therefore, the nets in $M \cup M'$ can not be routed without violating the second constraint on a CPL1-solution and hence π is not realizable. On the other hand, if at least one of w_{p_k} and $w_{p'_1}$ need not be a boundary wire, the nets in $M_{new} = M \cup M'$ are routable, which implies that M_{new} is a component. We call this merging of two parallel components a Type P *merging* and M and M' Type P *merging operands.* Either M or M' may be selected as the COA of the merging.

The type of the new component M_{new} depends on the types of M and M'. For example, as shown in Fig. 4 (a), when M is of *Type l* and M' is of *Type x*, M_{new} becomes a *Type l* component. Table 1 shows the component type of M_{new} for every possible combination of the component types of M and M', where $M \| M'$. There are four cases in which the merging is not possible due to a conflict of the boundary wires; in other words, the nets in $M \cup M'$ can not be routed without violating the second constraint on a CPL1-solution. We denote this by *'nil'* in the table and we say that the merging *fails.* Fig. 4 (b) shows such an example where M is of *Type r* and M' is of *Type l.*

The second merging operation is the merging of a component and two maximal clusters. Let M be a component defined above and assume that it is routalble. Let $C_1 = \{n_{u_i} = (t_{u_i}, b_{v_{m-i+1}}) \mid 1 \le i \le m,\ m < n\}$ and $C_2 = \{n_{u'_i} = (t_{u'_i}, b_{v'_{l-i+1}}) \mid 1 \le i \le l,\ l < n\}$ be two disjoint maximal clusters such that $[u_1, u_2, \ldots, u_m]$, $[v_1, v_2, \ldots, v_m]$, $[u'_1, u'_2, \ldots, u'_l]$ and $[v'_1, v'_2, \ldots, v'_l]$ are icseqs.

Lemma 1. *Suppose that* $u_m \oplus 1 = p_1$, $p_k \oplus 1 = u'_1$, $v'_l \oplus 1 = q_1$ *and* $q_k \oplus 1 = v_1$. *If* π *is realizable, the layout of* $C_1 \cup C_2$ *forms a* Type I *layout in any CPL1-solution.* □

Lemma 1 implies that $M_{new} = C_1 \cup M \cup C_2$ is a component. We call this merging of a component and two maximal clusters a Type X *merging* and C_1, M and C_2 Type X *merging operands.* In this case, the COA of the merging operation is the component M.

If the two maximal clusters C_1 and C_2 do not satisfy that $|C_2| - 1 \le |C_1| \le |C_2| + 1$, there is no CPL1-solution since $C_1 \cup C_2$ must form a Type I layout by Lemma 1 if π is realizable. Furthermore, if a necessary boundary wire of M conflicts with any wire for $C_1 \cup C_2$, π is not realizable. For example, if w_{p_1} must be a boundary wire and $|C_1| = |C_2| + 1$, then no CPL1-solution exists since both w_{p_1} and w_{u_m} have to pass between $b_{v'_l}$ and b_{q_1}.

Let $d = |C_1| - |C_2|$. The type of the new component M_{new} depends on the value of d and the component type of M. Table 2 shows the type of M_{new} for every possible combination of the value of d and the type of M. An example of a Type X merging is shown in Fig. 5 (a), where $d = -1$ and M is a *Type 0* component.

There are five cases for which a Type X merging fails, which is denoted by *'nil'* in the table. They are all due to a conflict between a boundary wire of M and some wire for $C_1 \cup C_2$ as explained above. It is easy to see that there is no CPL1-solution for each of those cases. Fig. 5 (b) shows such an example where $d = -1$ and M is a *Type r* component.

Lemma 1 holds even if C_1 (resp., C_2) is trivial and C_2 (resp., C_1) dose not exist. If π is realizable, C_1 (resp., C_2) forms a Type IM layout by itself and hence $M_{new} = C_1 \cup M$ (resp., $M \cup C_2$) becomes a component. We also call this merging of a component and a maximal cluster a Type X merging. Table 2 can still be used for this type of Type X merging to determine the type of a new component by assuming that $C_2 = \phi$ (resp., $C_1 = \phi$). Fig. 5 (c) shows an example of such a special Type X merging operation.

5. ALGORITHM DESCRIPTION

The algorithm consists of two main phases, the *merging phase* and *layout type assignment phase.* In the merging phase, the algorithm tests whether π is realizable using the two merging operations described in Section 4. If so, the algorithm proceeds to construct a CPL1-solution in the layout type assignment phase by specifying the layout type of each maximal cluster.

5.1 Merging Phase

The merging phase consists of the following two steps:

Step 1. Execute the following substeps until a merging operation fails or there remain no merging operands in CLIST.

a. Find Type P or Type X merging operands.
b. Perform the merging operation.

Step 2. If Step 1 results in a single component whose type is not *Type lr*, then π is realizable; go to the layout type assignment phase. Otherwise, terminate the algorithm with the answer that 'π is not realizable'.

We can find the first merging operands in Step 1 according to the following two lemmas.

Lemma 2. *If a given permutation π is realizable, then either $N(\pi)$ consists of a single cluster or it has at least two parallel maximal clusters.* □

Lemma 3. *Let C and C' be two maximal clusters. If $C \,||\, C'$, then each of the clusters is a component.* □

Thus, in Step 1 the algorithm first finds all maximal clusters in $N(\pi)$. This can easily be done by scanning the permutation backward from $\pi(n)$ to $\pi(1)$ and finding every maximal icseq.

If $N(\pi)$ is found to be a single cluster, then π is realizable and finding a CPL1-solution is trivial. We arbitrarily select a net and route it by a direct wire. The remaining nets can be routed in such a way that the layout is either a Type DL or Type DR layout.

While finding all maximal clusters in $N(\pi)$, the algorithm constructs a circular doubly linked list, called CLIST, which stores all the maximal clusters in the order of their appearances on C_{out}. The contents of CLIST will be changed during the execution of Step 1. Fig. 6 shows the initial status of CLIST for the instance in Fig. 2, where each maximal cluster is represented by a triangle with the mark 'c' in its inside.

If $N(\pi)$ consists of three or more maximal clusters, there must be two parallel maximal clusters if π is realizable. Otherwise, π is not realizable by Lemma 2 and hence the algorithm terminates.

We now assume that two parallel maximal clusters are found. Starting with any one of the two clusters as an initial COA, the algorithm repeatedly finds merging operands and performs a merging operation on them. If there are no merging operands including the current COA for a Type P or Type X merging operation, the next element in the clockwise direction in CLIST becomes a new COA. The merging operation is described below in detail.

When merging operands including the current COA are found, the algorithm tests whether they can be combined into a larger component which is routable. This is done by checking the corresponding item in Table 1 or Table 2 which is identified by the component types (and d in the case of a Type X merging) of the merging operands. If it is *nil*, π is not realizable due to the argument in Section 4 and the algorithm terminates. Otherwise, the algorithm produces a new component by combining the merging operands and assigns to it the value of the item in the merging table as its component type. Then M_{new} becomes a new COA and the merging operands are replaced by M_{new} in CLIST.

Since both parallel maximal clusters are components by Lemma 3, they form Type P merging operands. In this case, the algorithm assigns an appropriate component type to each maximal cluster, namely *Type 0* (resp., *Type x*) if it is trivial (resp., nontrivial), and performs a Type P merging operation. Note that the first merging operation which takes place in Step 1 is always this Type P merging of two parallel maximal clusters.

When a component M which consists of more than one maximal cluster is the current COA, the algorithm may sometimes find a maximal cluster C such that $C \,||\, M$ or $M \,||\, C$ in CLIST. In this case, M and C form Type P merging operands according to the following lemma.

Lemma 4. *Let M and C be a component and a maximal cluster, respectively, in $N(\pi)$. If $M \,||\, C$ or $C \,||\, M$, then C is a component.* □

Thus, the algorithm assigns an appropriate component type to C and performs a Type P merging operation.

Merging operations are iteratively performed until a merging operation fails for some merging operands or there remain no merging operands in CLIST. If a merging operation fails, it is clear that π is not realizable. Otherwise, the algorithm in Step 2 decides the existence of a CPL1-solution for π based on the following theorem.

Theorem 2. *π is realizable if and only if all maximal clusters in $N(\pi)$ are merged into a single component which is not of Type lr.* □

Note that if $N(\pi)$ is recognized to be a *Type lr* component by the algorithm, the two boundary wires for $N(\pi)$ have to pass between the same pair of adjacent terminals, which violates the second condition on a CPL1-solution.

During the merging phase, a directed graph called the *merging tree* is constructed. Initially, the graph has only isolated nodes which correspond to the maximal clusters. If a merging operation succeeds, the algorithm adds to the current graph a new node which corresponds to the resultant component and creates directed edges from this node to the nodes corresponding to the merging operands. Thus, if all the maximal clusters are eventually merged into a single component, the merging tree becomes a rooted directed tree. And if the root is not of *Type lr*, this tree will be used in the next phase to determine the layout types of the maximal clusters in $N(\pi)$.

We consider a simple example to illustrate some operations in the merging phase. Suppose that the algorithm is applied to the instance of Fig. 2 and $\{n_4, n_5\}$ is first selected as the COA of merging. Fig. 7 shows the merging tree and CLIST after the third merging operation. The final merging tree is shown in Fig. 8.

We now evaluate the time complexity for the merging phase in the following lemma and theorem.

Lemma 5. *If an element in CLIST becomes the COA for the second time, then there remain no merging operands in CLIST.* □

Theorem 3. *Given a permutation π of integers $1, 2, \ldots n$, the existence of a CPL1-solution for π can be tested in $O(n)$ time.*

Proof. It is clear that all the maximal clusters can be found in $O(n)$ time. Since the number of maximal clusters is at most n, both the construction of CLIST and finding a pair of parallel maximal clusters can be done in $O(n)$ time. In Step 1, if the algorithm can not find any merging operands including the current COA, it shifts the COA to the next element in CLIST in the clockwise direction. By Lemma 5, we need not move the COA more than n times in total for this case. Furthermore, the total number of the merging operations performed is at most $n-1$ because each merging operation reduces the number of elements in CLIST by at least one. This implies that the number of nodes in the merging tree is also $O(n)$. Since one execution of a Type P or Type X merging operation takes a constant time, all steps in the merging phase can be completed in $O(n)$ time. □

5.2 Layoput Type Assignment Phase

Once the merging phase is successfully completed, the layout type of each maximal cluster is to be determined. The merging tree is now a rooted directed tree. Its leaves correspond to the maximal clusters in $N(\pi)$ and the other nodes correspond to the components which have been found in the merging phase.

Starting from the root, the algorithm visits every node in the merging tree by Depth-First Search [1]. Suppose that the algorithm is currently visiting a node M_f. If M_f is the root which is a *Type x* component, its type is changed to either *Type l* or *Type r*. This selection is arbitrary.

If M_f is not a maximal cluster, it is obtained as the result of either a Type P or a Type X merging operation. In this case, the component type (or the layout type in the case of a maximal cluster) of each child of M_f is determined, according to the procedure *adjust* shown on the next page, so that the layout of the descendants of M_f is consistent with the component type of M_f. In the procedure, *ctype*$(\cdot)$ and *ltype*$(\cdot)$ are used to store the component and layout type, respectively, of a node and a single letter *i* (resp., J) is used to represent a *Type i* component (resp., Type J layout). Note that *ltype*$(\cdot)$ is used to describe only a Type I layout of one or two maximal clusters in the procedure. If the procedure is to assign a Type IM layout by itself to a trivial maximal cluster, then the algorithm specifies the location at which the corresponding wire has to pass.

If M_f is not only a component but also a maximal cluster, then its layout type is determined by its component type. If it is a trivial cluster, its layout is a Type D layout. Otherwise, its layout is a Type DL or Type DR layout depending on whether it is a *Type l* or *Type r* component. M_f can not be a *Type x* component when the algorithm visits this node because its type was already fixed when the algorithm visited its father and performed the procedure *adjust*.

Finally, if M_f is a maximal cluster but not a component, the algorithm performs no operation since M_f is one of Type X merging operands and hence its layout type was already determined when the algorithm visited its father.

We show examples of component type assignment for Type P and Type X merging operands in Figs. 9 and 10, respectively. It is assumed that the algorithm is currently visiting the father. In Fig. 9 (a), the type of M is changed from *Type x* to *Type l* since the father is of *Type l*. Since it is a maximal cluster, the algorithm assigns a Type DL layout to M when it visits M. In Fig. 9 (b), since the father is of *Type 0*, the type of M should be changed to *Type r* and its layout must be a Type DR layout. In Fig. 10 (a), the layout of $C_1 \cup C_2$ must be a Type IM layout and hence the type of M must be changed to *Type l*. In Fig. 10 (b), the trivial cluster C_1 has to form a Type IM layout by itself. It is easy to find that the wire w_4 must pass between b_2 and b_3. Because of this wire, the type of M has to be changed to *Type r*.

Lemma 6. *The layout type assignment phase can be completed in $O(n)$ time.*

Proof. As mentioned in the proof of Theorem 3, the number of nodes in the merging tree is $O(n)$.

```
procedure adjust
  begin
    if $M_f$ is obtained as the result of a Type P merging operation then
      \* Let $M$ and $M'$ be the two children of $M_f$ such that $M \| M'$. *\
      begin
        if $ctype(M) = x$ then if $ctype(M_f) = l$ or $lr$ then $ctype(M) \leftarrow l$ else $ctype(M) \leftarrow r$;
        if $ctype(M') = x$ then if $ctype(M_f) = 0$ or $l$ then $ctype(M') \leftarrow l$ else $ctype(M') \leftarrow r$
      end
    else if $M_f$ is obtained as the result of a Type X merging operation then
      \* Let $C_1, M$ and $C_2$ be the three children of $M_f$ defined in the Type X merging operation
         such that $C_1$ and $C_2$ are maximal clusters and $M$ is a component. *\
      begin
        if $d = 0$ then
          if $ctype(M_f) = l$ then begin
            $ltype(C_1 \cup C_2) \leftarrow$ IL; if $ctype(M) = x$ then $ctype(M) \leftarrow l$ end
          else begin $ltype(C_1 \cup C_2) \leftarrow$ IR; if $ctype(M) = x$ then $ctype(M) \leftarrow r$ end;
        if $d = 1$ then begin
          if $C_2 = \phi$ then $ltype(C_1) \leftarrow$ IM by itself else $ltype(C_1 \cup C_2) \leftarrow$ IM;
          if $ctype(M) = x$ then $ctype(M) \leftarrow r$ end;
        if $d = -1$ then begin
          if $C_1 = \phi$ then $ltype(C_2) \leftarrow$ IM by itself else $ltype(C_2 \cup C_1) \leftarrow$ IM;
          if $ctype(M) = x$ then $ctype(M) \leftarrow l$ end
      end
  end
```

Since the algorithm spends a constant time at each node as explained above, the layout type assignment phase requires $O(n)$ time in total. □

From Theorem 3 and Lemma 6, we finally arrive at our main theorem.

Theorem 4. *Given a permutation π of $\{1, 2, \ldots, n\}$, a CPL1-solution for π can be found, if one exists, in $O(n)$ time.* □

References

[1] A. V. Aho, J. E. Hopcraft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.

[2] S. Choe, T. Kashiwabara, and T. Fujisawa, "Restricted Permutation Layout," *Trans. of the IECE of Japan*, vol. J68E, pp. 269–276, 1985.

[3] S. Choe, T. Kashiwabara, and T. Fujisawa, "A Permutation Layout with Limited Between-pin Congestion," *Papers of Technical Group on Circuit and Systems of the IECE of Japan*, CAS 83–74, 1983.

[4] M. Cutler and Y. Shiloach, "Permutation Layout," *Networks*, vol. 8, pp. 253–278, 1978.

[5] T. Kashiwabara, K. Itagaki, S. Masuda, and T. Fujisawa, "On Certain Permutation Layout," *Proc. 1985 IEEE Int. Symp. on Circuits and Systems, Kyoto,Japan*, June 1985, pp. 1043–1046.

[6] T. Ozawa, "A Routing Procedure for an IC Module with Many Pins - A Solution to a Circular Permutation Layout Problem," *Networks*, vol. 15, pp. 33–48, 1985.

[7] C. S. Rim, N. J. Naclerio, S. Masuda, and K. Nakajima, "A Linear Time Algorithm for Circular Permutation Layout," Technical Report, Systems Research Center, Univ. of Maryland, College Park, MD, 1987.

[8] I. Shirakawa, "Some Comments on Permutation Layout," *Networks*, vol. 10, pp. 179–182, 1980.

[9] S. Tsukiyama and E. S. Kuh, "Double-Row Planar Routing and Permutation Layout," *Networks*, vol. 12, pp. 287–316, 1982.

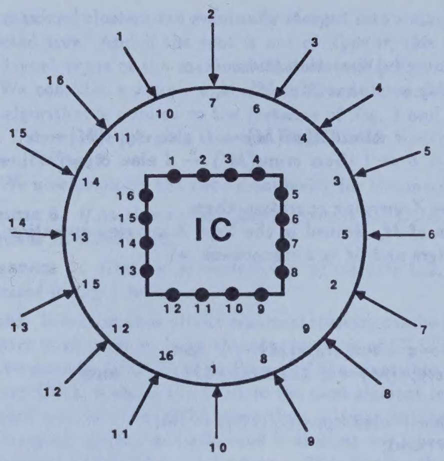

IC
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Fig. 1. Application of circular permutation layout in a single layer routing of an IC (module).

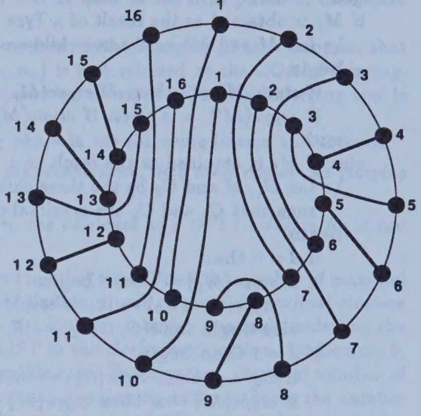
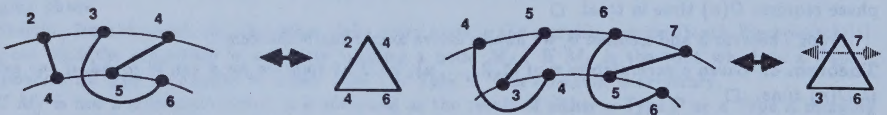
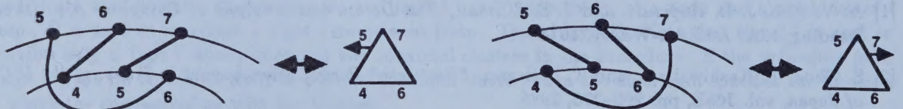

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Fig. 2. A CPL1-solution for the permutation $\pi =$ 10 7 6 4 3 5 2 9 8 1 16 12 15 13 14 11.


2
3
4
4
5
6
2 4
4 6
4
5
6
7
3
4
5
6
4 7
3 6

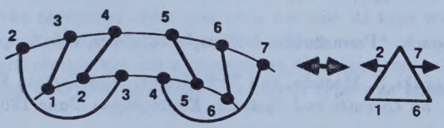
(a) *Type 0* component.

(b) *Type x* component.


5
6
7
4
5
6
5 7
4 6
5
6
7
4
5
6
5 7
4 6

(c) *Type l* component.

(d) *Type r* component.


2
3
4
5
6
7
1
2
3
4
5
6
2 7
1 6

(e) *Type lr* component.

Fig. 3. Typical layouts for component types and their symbols (An arrow in a symbol indicates a layout of a component needs a boundary wire on that side).

Table 1. Type P merging table.

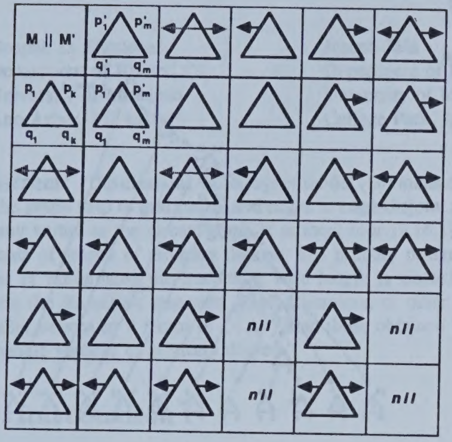
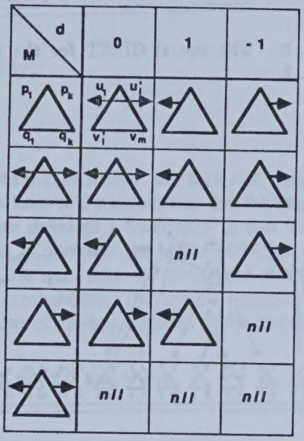
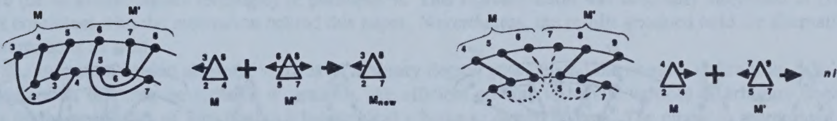

M || M'
p_1' p_m' q_1' q_m'
p_1 p_k q_1 q_k
nil
nil
nil
nil

Table 2. Type X merging table.

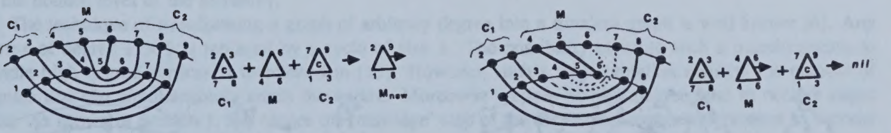

d
M
0
1
-1
p_1 p_k q_1 q_k
u_1 u_l' v_1' v_m
nil
nil
nil
nil
nil


M
M'
M
M'
M_{new}
nil

(a) A successful Type P merging.

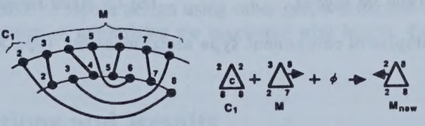
(b) A failed Type P merging.

Fig. 4. Examples of Type P merging operations.


C_1
M
C_2
C_1
M
C_2
M_{new}
nil

(a) A successful Type X merging, where $d = -1$ and M is a *Type 0* component.

(b) A failed Type X merging, where $d = -1$ and M is a *Type l* component.


M
C_1
C_1
M
ϕ
M_{new}

(c) A successful Type X merging when $C_2 = \phi$, $d = 1$, and M is a *Type l* component.

Fig. 5. Examples of Type X merging operations.

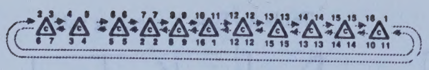


Fig. 6. The initial CLIST for the instance in Fig. 2.

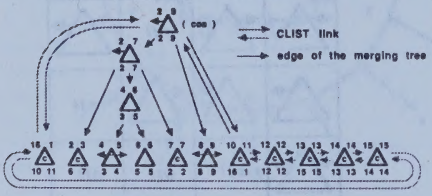

(cos)
CLIST link
edge of the merging tree

Fig. 7. Intermediate merging tree and CLIST after the third merging operation.

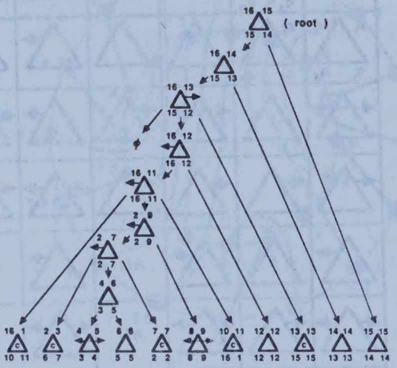
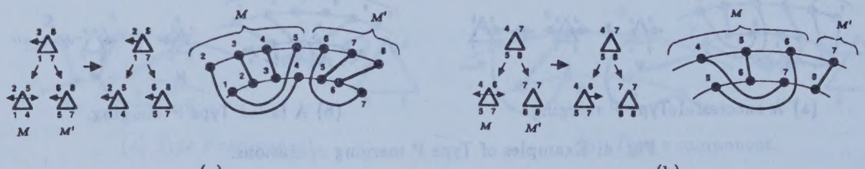


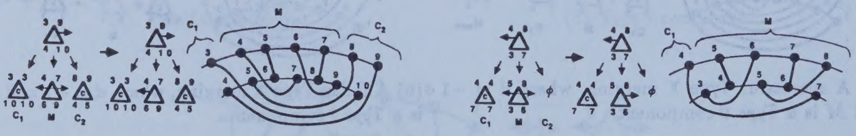
Fig. 8. Result of merging operations for the instance of Fig. 2.



(a)

(b)

Fig. 9. Examples of component type assignment for Type P merging.


C1
M
C2
C1
M
C2
C1
M
C2
φ
φ
C1
M

(a) $C_1 \cup C_2$ forms a Type IM layout.

(b) C_1 itself forms a Type IM layout.

Fig. 10. Examples of component type assignment for Type X merging.

Layouts of Graphs of Arbitrary Degree

Deepak D. Sherlekar
Department of EE and CS
University of Michigan
Ann Arbor, MI 48109

Joseph JáJá
Department of Electrical Engg.
University of Maryland
College Park, MD 20742

Abstract: The classical VLSI layout model [14] maps circuit graphs into an integer grid such that vertices of the graph map to grid points and edges to edge-disjoint paths. This leads to the assumption that the degree of any vertex in the circuit graph is at most four [9,16,1]. This paper develops a framework to find VLSI layouts of graphs of arbitrary degree. The primary design concern is to minimise the layout area , but the issue of minimising the maximum wire length is treated as well. The solutions use divide-and-conquer, where the dissection step uses graph separators, or other dissection mechanisms. The layouts obtained are smaller in area by a factor of $O(\Delta^2)$ than those obtained by extensions of existing schemes in the literature to handle vertices of arbitrary degree.

1 Introduction

An area-efficient scheme to find VLSI layouts for graphs of arbitrary degree is presented. Existing schemes assume graphs of degree at most 4, since they adopt the classical layout model [14], where vertices map to points on an integer grid, and edges map to edge disjoint paths. This constraint can relaxed by extending the grid model to represent vertices of high degree. This is done by representing a vertex of degree d by a square (or an almost-square rectangle) of perimeter d. This representation was originally suggested in [14], and is consistent with the motivation behind this paper. Nevertheless, the results obtained hold for alternative representations as well.

The motivation behind allowing vertices of arbitrary degree goes beyond the obvious desirability to relax an assumption that may be violated in practice. An efficient scheme to handle vertices of arbitrary degree opens up the possibility of formulating a hierarchical scheme to design layouts. The circuit is represented as a family of graphs corresponding to this hierarchy. The graph at the bottom level is the circuit graph, whose vertices correspond to components and edges to wires connecting them. At other levels, vertices correspond to modules, instead of elementary circuit elements. This design strategy is especially useful when the circuit has also been designed hierarchically, and is composed of sub-circuits which already possess efficient layouts. Under the above scenario, a vertex representing a module may have a large number of edges incident with it. Hence the assumption of vertices of degree at most 4 does not hold, regardless of whether or not it holds at the bottom level of the hierarchy.

The technique of transforming a graph of arbitrary degree into a trivalent graph is well known [6]. Any vertex of degree $k > 3$ is replaced by a cycle of size k. The possibility of using such a transformation to handle high degree vertices is mentioned in [10]. However, such a transformation restricts the number of signals that can simultaneously reach the vertex. Moreover, vertices of high degree tend to occupy larger area. As shown in Section 3, this causes the 'marriage' step of the divide-and-conquer algorithms to become very expensive.

Section 2 defines the problems solved and describes the results. Section 3 discusses the problems posed by vertices of high degree, and proposes a solution strategy based on binary tree embeddings. Layouts of n^α-separable graphs, $0 < \alpha < 1$, and 1-separable graphs are described in Sections 4 and 5 respectively. Section 6 addresses the problem of layout design using other graph dissection mechanisms such as the one proposed in [7], and the problem of minimising the maximum wire length. Concluding remarks appear in Section 7.

2 Problem Definitions and Results

The purpose of this paper is to develop a framework to find VLSI layouts of graphs of arbitrary degree. The basic scheme involves deriving a layout through an embedding of the given graph in a binary tree. For the layouts to be compact, the embedding needs to satisfy special properties. These are met by defining a property called the *external edge balancing condition*, which must be satisfied by the separators used to find the embedding. An algorithm is presented to compute such separators from conventional separators.

Dissection Mechanism	Maximum Degree (Δ)	Layout Area	Reference
1-separator,	4	$O(n)$	[10]
	PARAMETER	$O(\Delta n \log \Delta)$	*Theorem 5.1*
n^α-separator, $\alpha < 1/2$	4	$O(n)$	[10]
	PARAMETER	$O(\Delta^{1/1-\alpha} n)$	*Theorem 4.4*
n^α-separator, $\alpha = 1/2$	4	$O(n \log^2 n)$	[10]
	PARAMETER	$O\left(\Delta^2 n \log^2 \frac{n}{\Delta^2}\right)$	*Theorem 4.4*
n^α-separator, $\alpha > 1/2$	4	$O(n^{2\alpha})$	[10]
	PARAMETER	$O(\Delta^2 n^{2\alpha})$	*Theorem 4.4*
$(F, \sqrt{2})$ bifurcator	4	$O\left(F^2 \log^2 \frac{n}{F}\right)$	[1]
$(F, \sqrt{2})$ weighted bifurcator	PARAMETER	$O\left(F^2 \log^2 \frac{n}{F}\right)$	*Theorem 6.2*

Table 1: Layouts of Different Classes of Graphs

The resulting bounds on layout area are better by a factor of $O(\Delta^2)$ than those achieved by extending any of the existing techniques to handle graphs of arbitrary degree. Moreover, these bounds match those in the literature for bounded degree graphs having identical edge separators. These results are summarised in Table 1 along with corresponding results in the literature for graphs of maximum degree 4.

It is observed that extensions of existing techniques based on *bifurcators* and the *tree of meshes* [1] to handle graphs of arbitrary degree are also inefficient. For graphs having an $(F, \sqrt{2})$ bifurcator, the layout area for graphs of maximum degree 4 is $O(F^2 \log^2(n/F))$ [1]. Extending the scheme to handle vertices of maximum degree Δ results in layout areas which are larger by a factor of $O(\Delta^2)$. A modification to the framework in [1] is proposed to remedy this shortcoming. This is done by introducing *weighted* bifurcators, and computing *weighted* bifurcators satisfying the *external edge balancing condition.* For $(F, \sqrt{2})$ weighted bifurcators, the resulting technique yields layouts whose area is $O(F^2 \log^2(n/F))$, regardless of the maximum degree. The layouts obtained are also modified to minimise the maximum wire length while limiting the increase in area to a constant factor.

3 A Strategy Based on Tree Embeddings

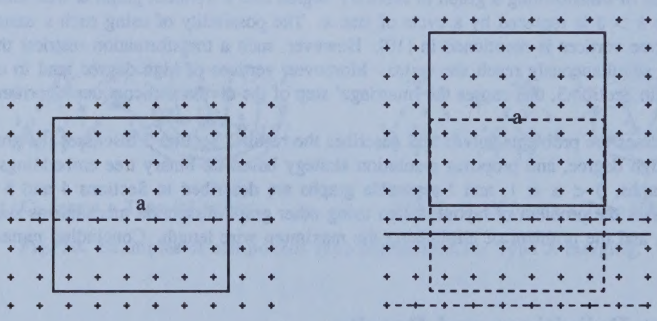


Figure 1: Channel Creation in the Presence of Large Vertices

The large size of high degree vertices makes the *marriage* step of the layout algorithms in [10] and [16] unacceptably expensive. Consider the process to reinsert an edge deleted during the *dissection* phase of the schemes in [10,16]. Figure 1 demonstrates the creation of a channel for an edge in the presence of vertices of high degree. The process of 'sweeping off' vertices that lie in the path of the edge may increase the dimensions of the two layouts considerably. For graphs of maximum degree Δ, the increase in the width and height of the layout for every inserted edge can be as high as $O(\Delta)$. The resulting layout area will be larger by a factor of $O(\Delta^2)$ than that of the corresponding bounded degree graph having the same edge separator. The layout strategy presented in this paper demonstrates that this gap can be bridged completely in most cases, and reduced to a slowly growing function of Δ in others.

3.1 Outline of the Strategy

Graph embeddings have been suggested as a means to solve many difficult graph problems [12,11,13]. The approach involves solving a problem on a complex graph by making it 'look like' one that is easier to analyse and manipulate. The former is called the source graph, and the latter, the target graph. This involves embedding the source graph in the target graph such that the relevant information in the source graph is preserved as well as possible. With reference to the VLSI layout problem, binary trees are a natural choice as a target graph. The *H-Tree* scheme [3] produces optimal layouts for complete binary trees. Thus the graph is first embedded in a binary tree. The original identity of the graph is then restored by enlarging the vertices and edges of the binary tree. The resulting structure contains all the information of the original graph, and has a shape that resembles a binary tree.

The *H-Tree* layout scheme was initially proposed for layouts of complete binary trees. However, not all properties of the complete binary tree are crucial to the success of the method. For the method to work well, one only need ensure that: (1) the tree is balanced, (2) the vertices are rectangles of small aspect ratio, (3) all edges at the same level have approximately equal thickness and (4) all vertices at the same level have approximately equal size.

If the structure resulting from the enlarging process satisfies all the conditions mentioned above, it is called a *Broad Band Tree*. The sizes of the enlarged vertices in a *Broad Band Tree* will typically reduce exponentially as one goes down the tree. The *attrition factor* of the *Broad Band Tree* is the rate at which the sides of the vertices in the tree shrink as one goes down the tree.

For graphs satisfying vertex separators, an embedding that corresponds naturally to the recursive decomposition of the graph by its separator is appropriate. A vertex in the binary tree is associated with *all* the vertices of the separator corresponding to its level and position in the recursive decomposition of the graph. The recursion bottoms out upon encountering completely disconnected graphs, which are mapped in the leaves. Strong separators (i.e., those which produce equal partitions) are used to ensure that the resulting tree is balanced.

3.2 Edge and Vertex Enlargements

The process of restoring the original identity of the graph starts at the root and operates breadth first on the binary tree. Every vertex of the binary tree is replaced by a *supervertex*. This *supervertex* is a VLSI layout which corresponds to the separator mapped in the vertex, and the wires of the original graph that pass through it. An edge is replaced by a bundle of wires. These wires correspond to the edges that connect the subgraph mapped beneath the edge to the rest of the graph. The *marriage phase* of the layout algorithms in [10,16] is expensive because channels for wires connecting two layouts are cut through their entire width and height. In contrast, all wires in the method proposed here are routed along the edges of the binary tree. At any vertex, the wires reaching it from above are of two types. One consists of the wires incident with the separator mapped in the vertex, and the other of the wires that are being routed through it.

Let v be the root of any subtree T_v of the binary tree. Let G_v be the the subgraph mapped in T_v. The separator S_v partitions G_v into G_{v0} and G_{v1}. S_v is mapped in v, G_{v0} in the subtree T_{v0} of T_v, and G_{v1} in the subtree T_{v1} of T_v. Then the *supervertex* replacing v is generated by the following prescription:

Find a VLSI layout of the subgraph induced by S_v and all edges having an end point in S_v. The layout is similar to that used for a separator in Valiant's method [16], and is produced by assigning a horizontal and a vertical channel to every edge. The edges handled by the supervertex replacing v are of two types: those coming from above, and those which are deleted when the separator S_v splits G_v into G_{v0} and G_{v1}. Those edges which come from above and are incident with S_v are connected to it. The rest of them are laid vertically alongside the layout of S_v. These edges, along with the edges extending downward from S_v, are incident with G_{v0} and G_{v1}. These two groups of edges enter a switch which splits them into two bundles, one incident withG_{v0}, and the other with G_{v1}. These bundles are routed appropriately through the edges linking v to its children. The area of the switch is quadratic in the number of edges it handles. Figure 2 illustrates the resulting supervertex which replaces v.

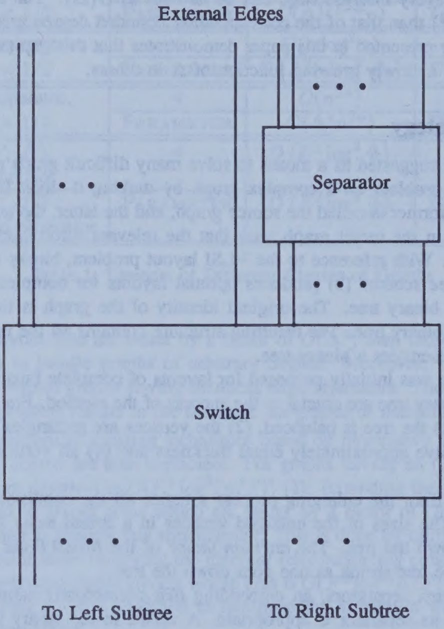


Figure 2: Details of a Broad Band Tree Supervertex

3.3 Ensuring Balanced Enlargements

The scheme described thus far does not guarantee vertices and edges of approximately equal size at the same level. Consider the factors affecting the edge and vertex enlargements. The thickness of an edge depends on the number of graph edges passing through it. The area of a *supervertex* varies quadratically with two parameters. The first is the total number of edges incident with at least one vertex of the separator, and the second is the number of edges routed through v by its parent. If *strong f(n)-separators* are used, the bound on the size of a separator mapped to any vertex v at level i is $O(f(n/2^i))$. Hence the bound on the number of edges incident with these vertices is $O(\Delta f(n/2^i))$, where Δ is the maximum vertex degree in the graph. However, the number of edges being routed to v from its parent may vary widely. This depends on the portion of edges routed through the parent of v which are also routed through v. These edges are essentially the edges of attachment of the subgraph mapped in the tree rooted at v. If the *strong separator* used also partitions the external edges of attachment of a graph nearly equally over its two partitions, the resulting enlargement can be guaranteed to be balanced. This condition will be referred to as the *external edge balancing condition.* The structure resulting from this balanced enlargement yields a *Broad Band Tree* having a root *supervertex* of size $O(\Delta f(n)) \times O(\Delta f(n))$ for *strong f(n)-separators* satisfying the *external edge balancing condition.*

A *strong separator* satisfying the *external edge balancing condition* may be looked upon as a separator that partitions a graph with respect to two kinds of weights: one corresponding to the vertices, and the other to the external edges. Such a separator was introduced in [5] for the class of chordal graphs. The paper also mentions that a similar separator for planar graphs appears in Gilbert's thesis [4]. Given a graph with two independent sets of vertex weights, each summing to no more than one, a *two-weight* separator splits it in two parts, each having at most half the weight of the first kind, and a weight of at most $((1+\epsilon)/2)$ of the second kind, for some fixed $\epsilon < 1$. If a graph possesses a *two-weight* $f(n)$-separator theorem, then a *strong* $f(n)$-separator theorem satisfying the *external edge balancing condition* can be derived easily. Assign a weight of $1/n$ of the first kind to every vertex. If the graph has a total of m_e external edges, then for any vertex incident with m_i external edges, assign a weight of the second kind of m_i/m_e. Then the *two-weight* separator will partition the graph into two parts, each having at most $n/2$ vertices, and at most $m_e(1+\epsilon)/2$ external edges.

4 Layouts of n^α - Separable Graphs

Following the approach outlined in the previous section, the first step is to derive a *two-weight* n^α-separator theorem for any graph having an n^α-separator theorem.

Lemma 4.1 *Any n^α-separable graph, $0 < \alpha < 1$, also possesses a* two-weight *n^α-separator theorem.*

Proof : It is known that n^α-separable graphs are also *strongly* n^α-separable [15]. Let $\nu = \lceil 1/\epsilon \rceil$. Divide the graph into ν parts, each of weight at most ϵ of the second kind by applying the *strong* n^α-separator recursively $\log \nu$ times. This requires

$$O\left(\sum_{i=0}^{\log \nu - 1} 2^i (n/2)^{i\alpha}\right) = O(n^\alpha)$$

vertices. Now apply the *strong* n^α-separator to each of these ν partitions, splitting them into two equal parts by weight of the first kind. This also requires at most $O(n^\alpha)$ vertices. Now there are ν pairs of graphs, $< G_{11}, G_{12} >, \ldots, < G_{\nu 1}, G_{\nu 2} >$. Form the two partitions desired, each consisting of a union of ν of these graphs, one from every pair. This is done while minimising the difference in weight of the second kind between the two partitions. Thus each partition has at most half the weight of the first kind, and at most $(1+\epsilon)/2$ weight of the second kind. □

This leads to the following theorem.

Theorem 4.2 *An n^α-separable graph, $0 < \alpha < 1$, also possesses an n^α-separator theorem satisfying the* external edge balancing condition. □

Although these results ensure that the external edges are divided nearly equally over the two partitions, the choice of a proper value for ϵ is crucial. This shall soon become clear when this separator theorem is applied to derive the *Broad Band Trees* and layouts.

It will be shown that a *Broad Band Tree* obtained using n^α-separators satisfying the *external edge balancing condition* has an *attrition factor* of 2^α. The following lemma gives the layout of such a *Broad Band Tree* when its root *supervertex* is of size $p \times p$.

Lemma 4.3 *The layout area of a* Broad Band Trees *having a root* supervertex *of size $p \times p$,* attrition factor *2^α, $0 < \alpha < 1$, and depth $2l$ is*

$$\begin{array}{lll} O(p^2) & \text{if} & \alpha > 1/2 \\ O(p^2 l^2) & \text{if} & \alpha = 1/2 \\ O(p^2 2^{2l(1-2\alpha)}) & \text{if} & \alpha < 1/2 \end{array}$$

Proof : Apply the *H-Tree* scheme [3] to find the layout of the *Broad Band Tree*. Notice that the vertices no longer have unit area, or the edges, unit thickness. Every set of 'Hs' introduced accounts for 2 levels of the *Broad Band Tree*. Thus the recurrence H_α defining the height of the layout is:

$$\begin{aligned} H_\alpha(p, 2l) &= 2H_\alpha(p/2^{2\alpha}, 2l-2) + p \\ H_\alpha(p, 0) &= p \end{aligned}$$

Thus

$$H_\alpha(p, 2l) = \sum_{i=0}^{l} 2^i (p/2^{2i\alpha})$$

Hence

$$H_\alpha(p, 2l) = \begin{cases} \Theta(p) & \text{if } \alpha > 1/2 \\ p(l+1) & \text{if } \alpha = 1/2 \\ \Theta(2^{(1-2\alpha)l} p) & \text{if } \alpha < 1/2 \end{cases}$$

A recurrence W_α for the width of the layout may be formulated similarly. It can be shown that W_α is of the same order as H_α. The result follows. □

The following theorem finds the layout area of graphs having an n^α-separator theorem.

Theorem 4.4 *Any n^α-separable graph, $0 < \alpha < 1$, having n vertices, $O(n)$ edges, and maximum degree Δ can be laid out in area*

$$\begin{array}{ll} O(\Delta^2 n^{2\alpha}) & \text{if} \quad \alpha > 1/2 \\ O\left(\Delta^2 n \log^2(n/\Delta^2)\right) & \text{if} \quad \alpha = 1/2 \\ O\left(n\Delta^{1/(1-\alpha)}\right) & \text{if} \quad \alpha < 1/2 \end{array}$$

Proof : Apply the n^α separator satisfying the *external edge balancing condition* to derive the *Broad Band Tree* for the graph. From the discussion in Section 3.3, the size of any supervertex in the *Broad Band Tree* is affected by two factors. The first is the number of edges incident with the separator vertices, and the second is the number of external edges routed through the supervertex.

The separator produces partitions which are at most half the size of the graph. So the size of the separators at level i of the recursion is $O((n/2^i)^\alpha)$. Thus the number of edges incident with any separator at level i is at most $O(\Delta(n/2^i)^\alpha)$. At any level, the edges external to the subgraph being bisected get split over its two partitions such that each partition has at most $(1+\epsilon)/2$ of them, where ϵ is the constant in the two-weight separator of Lemma 4.1. Thus the total number of edges handled by the supervertices at different levels of the *Broad Band Tree* is:

$$\begin{array}{lll} level\ 0: & & c\Delta n^\alpha \\ level\ 1: & & c\Delta n^\alpha(1+\epsilon)/2 + c\Delta(n/2)^\alpha \\ & \vdots & \\ level\ i: & & c\Delta \displaystyle\sum_{j=0}^{i} \left(\frac{1+\epsilon}{2}\right)^j \left(\frac{n}{2^{i-j}}\right)^\alpha \\ & = & c\Delta \left(\dfrac{n}{2^i}\right)^\alpha \displaystyle\sum_{j=0}^{i} \left(\frac{1+\epsilon}{2^{1-\alpha}}\right)^j \\ & = & O\left(\Delta(n/2^i)^\alpha\right) \end{array}$$

provided $1+\epsilon < 2^{1-\alpha}$, i.e., $\epsilon < 2^{1-\alpha} - 1$. Thus the choice of *epsilon* is crucial in ensuring a small layout area. Choose an ϵ satisfying this inequality. This yields a *Broad Band Tree* having a root supervertex of size $O(\Delta n^\alpha) \times O(\Delta n^\alpha)$. The *attrition factor* of the *Broad Band Tree* is 2^α, and the height is the level L at which all edges are deleted. Thus the value of L is given by:

$$L = \log(n/\Delta^{1/(1-\alpha)}) + \Theta(1)$$

The layout of the *Broad Band Tree* above is the layout of the graph. The desired result may be shown by applying Lemma 4.3 followed by some algebraic simplification. □

5 Layouts of 1 - Separable Graphs

It can be shown that 1-separable graphs are *strongly* $\log n$-separable [15]. This in turn, implies a *strong* n^α-separator for any fixed $\alpha > 0$. Using Theorem 4.4, it can be shown that any 1-separable graph of maximum degree Δ has a layout of area $O(n\Delta^{1+\epsilon})$ for any fixed $\epsilon > 0$. However, it is possible to do better.

Theorem 5.1 *Any 1-separable graph of maximum degree Δ has a layout of area $O(n\Delta \log \Delta)$.*

Proof : It is possible to derive a two-weight $\log n$ separator theorem for any 1-separable graph on the same lines as the proof of Lemma 4.1. This in turn implies a *strong* $\log n$-separator satisfying the *external edge balancing condition.* Proceed along the lines of the proof of Theorem 4.4. Recall the constant ϵ which determines the precision with which the two-weight separator splits the graph with respect to the second weight. It turns out that any $\epsilon > 0$ is adequate to ensure that the size of the supervertices is not dominated by the number of external edges passing through it.

The recurrence for the height H_1 of the layout is:

$$\begin{aligned} H_1(c\Delta \log n, L) &= 2H_1(c\Delta(\log n - 2), L-2) + c\Delta \log n \\ H_1(s,0) &= s \end{aligned}$$

where L is the level of recursion at which all edges are deleted, and c is some fixed constant.

Solving for H_1,

$$\begin{aligned} H_1(c\Delta \log n, L) &= c\Delta \sum_{i=0}^{\lceil L/2 \rceil} 2^i (\log n - 2i) \\ &= O\left(\Delta \sum_{i=0}^{L} 2^{i/2}(\log n - i)\right) \\ &= O\left(\Delta 2^{L/2}(\log n - L + \Theta(1))\right) \end{aligned}$$

As before, the value of L may be shown to be $\log(n/(\Delta \log \Delta)) + \Theta(1)$. Hence the height of the layout is given by

$$H_1(c\Delta \log n, L) = O\left(\Delta \sqrt{\frac{n \log \Delta}{\Delta}}\right) = O\left(\sqrt{n\Delta \log \Delta}\right)$$

It can be shown similarly that the width $W_1(c\Delta \log n, L) = O(\sqrt{n\Delta \log \Delta})$.

Hence the area of the layout is $O(n\Delta \log \Delta)$ □

The above result can be easily generalised to handle $\log^k n$ separable graphs, for any fixed $k \geq 0$. The result is stated in the following theorem, whose proof is omitted.

Theorem 5.2 *Any* $\log^k n$*-separable graph of maximum degree* Δ *has a layout of area* $O(n\Delta \log^{k+1} \Delta)$. □

6 Alternative Dissection Mechanisms

From Sections 4 and 5, it is clear that extensions of existing methods yield layouts whose area is larger by a factor of $O(\Delta^2)$ than those obtained by the methods here. The bound attained by the bifurcator-based approach in [1] is also worse by a factor of $O(\Delta^2)$. However, the approach resembles that presented here. It was shown in [7] that separators are incapable of accurately predicting the actual number of edges required to partition a graph at every level of the recursive decomposition. Bifurcators were introduced as an alternative dissection mechanism to overcome this deficiency. Thus it would be useful to examine the possibility of modifying this approach to obtain area-efficient layouts. The problem encountered by bifurcators in handling vertices of high degree is examined below in view of the results obtained earlier in this paper.

6.1 The Bifurcators/Tree of Meshes Framework

A graph is said to have a $(F, \sqrt{2})$*-bifurcator*, or simply an F*-bifurcator* [7] if it can be partitioned into roughly equal parts by removing F edges, the resulting partitions by removing $F/\sqrt{2}$ edges, the partitions after 2 bisections by removing $F/\sqrt{2^2}$ edges, and so on. If the partitions produced are equal, it is called a *balanced bifurcator*.

In [1], Bhatt and Leighton presented a new framework, based on bifurcators, for finding VLSI layouts. Their approach resembles the scheme presented here in that the layouts are obtained through an embedding into an intermediate structure, whose layout, in turn, yields a layout of the graph. The intermediate structure in [1] is the *tree of meshes.* The structure was proposed in an earlier paper [8] to prove a lower bound of $\Omega(n \log n)$ on the layout area of planar graphs. However, unlike the scheme described here, the intermediate structure requires no further transformation. In fact, the *tree of meshes* is similar in shape to the *broad band tree.*

The scheme in [1] also requires the bifurcators to satisfy a balancing property concerning external edges. Bifurcators satisfying this property are called *fully balanced bifurcators.* However, for graphs of maximum degree Δ, the size of the fully balanced bifurcator is larger by a factor of $O(\Delta)$. This leads to layouts whose resulting area is larger by a factor of $O(\Delta^2)$ than corresponding graphs of maximum degree 4.

The approach developed here can be used in conjunction with bifurcators to derive the corresponding tree of meshes in a natural way by starting with an embedding into a binary tree. The embedding maps completely disconnected graphs at the bottom level of the recursive decomposition into the leaves of the binary tree. No vertices are mapped to the interior vertices of the tree. For $(F, \sqrt{2})$ *fully balanced bifurcators,* the resulting *Broad Band Tree* has root supervertex of $O(F) \times O(F)$, and *attrition factor* of $\sqrt{2}$. This is essentially a *tree of meshes* $T_{F,2\log n/F}$.

6.2 Modifying Bifurcators for High Degree Vertices

The *external edge balancing condition* is inherently weaker than the condition met by *fully balanced* bifurcators. The *fully balanced bifurcator* guarantees that for all i, $1 \leq i \leq \Delta$, vertices having i previously deleted edges are distributed equally over the two partitions. Unlike *fully balanced bifurcators,* the *external edge balancing condition* is a *global* control of the distribution of previously deleted edges over the partitions. Moreover, the difference between the number of external edges incident with the two partitions is only $O(\Delta^2)$ for *fully balanced bifurcators,* but can be up to a constant factor of the total number of external edges for separators satisfying the *external edge balancing condition.* These observations may be used to modify the layout methodology based on bifurcators to handle vertices of high degree. To start with, define a *weighted bifurcator*:

Consider a graph G having nonnegative weights on its vertices which sum to no more than one. G has a *weighted* F bifurcator if

- G can be split into two partitions, each of weight at most $2/3$, by removing F edges.
- Each of the partitions of G has a *weighted* $F/\sqrt{2}$ bifurcator.

A *weighted* $(F, \sqrt{2})$ bifurcator satisfying the *external edge balancing condition* is defined in a manner analogous to that of separators.

Theorem 6.1 *Any graph having a weighted F-bifurcator also has a weighted F-bifurcator satisfying the* external edge balancing condition.

Proof : The weighted bifurcator may be used to compute a two-weight F-bifurcator exactly as in Lemma 4.1. If the graph has two sets of vertex weights, each summing to no more than 1, then a two-weight bifurcator will split it into two parts, each having at most $1/2$ weight of the first kind, and at most $(1+\epsilon)/2$ weight of the second kind. Moreover, each partition will recursively satisfy a two-weight $F/\sqrt{2}$ bifurcator. Associate the first weight with the number of vertices, and the second with the number of external edges. The result follows. □

The above theorem directly leads to a strategy to compute layouts of arbitrary degree graphs having a weighted F bifurcator for any F.

Theorem 6.2 *Any graph having a $(F, \sqrt{2})$ weighted bifurcator has a layout of area $O(F^2 \log^2(n/F))$ regardless of its maximum vertex degree.*

Proof : Apply the same same method as that used for n^α-separable graphs with the following modifications. Embed the vertices of the completely disconnected graphs at the bottom level of the recursive decomposition into the leaves of the binary tree. This yields a balanced binary tree of depth $(2\log(n/F) + \Theta(1))$. Obtain the corresponding *Broad Band Tree* following the procedure outlined in Section 3.2. Following the argument in Theorem 4.4, the value of ϵ in the two-weight bifurcator is chosen to be $\sqrt{2} - 1$. The result is a *Broad Band Tree* having root supervertex of side $O(F)$, attrition factor $\sqrt{2}$, and depth $2\log(n/F) + \Theta(1)$. The area of the *Broad Band Tree* is $O(F^2 \log^2(n/F))$. □

6.3 Minimising the Maximum Wire Length

An important parameter affecting the speed of a chip is the length of the longest wire. This problem has been addressed by [2] and [1]. From the preceding discussion in this section, it is clear that the *Broad Band Tree* structure is similar to the *Tree of Meshes*. Hence the technique in [2] to minimise the maximum wire length can be adapted to the layouts for graphs of arbitrary degree as well.

The wire length is reduced in two stages. The first stage attempts to minimise the number of supervertices spanned by an edge, and the second phase tries to minimise the maximum wire length between supervertices. The second stage is exactly the same as that described in [1]. The statement of that result in terms of *Broad Band Trees* having root supervertex of size $O(F) \times O(F)$ and *attrition factor* $\sqrt{2}$ is given below.

Lemma 6.3 *(Adapted from [1]) A* Broad band Tree *with an* $O(F) \times O(F)$ *root supervertex and* attrition factor $\sqrt{2}$ *has a layout of area* $O(F^2 \log^2(n/F))$, *such that the length of any edge connecting two supervertices is at most* $O(F \log(n/F)/\log\log(n/F))$. □

The first stage starts at the root and proceeds downwards, applying a transformation to all supervertices at the same level. All vertices of the graph which are incident with deleted edges and are embedded at lower levels are 'pulled up' and embedded in the supervertex being transformed. All remaining edges incident with these vertices are also deleted. If the transformation is applied at every k levels, the number of supervertices spanned by an edge will be limited to at most k. The choice of k influences the effectiveness of the second stage. If the vertices are pulled in too frequently, the sizes of the supervertices will grow too large, thus affecting the second stage. Following an analysis similar to that in [1], the area of the resulting *Broad Band Tree* increases by a factor of $O(\Delta^2)$ if $k = \log \Delta$, and by a constant factor if $k = \Delta$. In the former case, the maximum distance between supervertices is $O(F\Delta \log(n/\Delta F)/\log\log(n/\Delta F))$. In the latter case, it is $O(F \log(n/F)/\log\log(n/F))$. Thus it is better to use $k = \Delta$. This leads to the following theorem.

Theorem 6.4 *Let G be a graph having a weighted* $(F, \sqrt{2})$ *bifurcator and maximum degree* Δ. *Then G has a layout of area* $O(F^2 \log^2(n/F))$ *and maximum wire length* $O(F\Delta \log(n/F)/\log\log(n/F))$. □

The above technique can be applied to other separable graphs as well. Any graph G having a $\sqrt{n}$-separator theorem and maximum degree Δ has a weighted $\left(O(\Delta\sqrt{n}), \sqrt{2}\right)$ bifurcator. Hence from the above theorem, G has a layout of area $O(\Delta^2 n \log^2(n/\Delta^2))$, such that the maximum wire length in the layout is at most $O(\Delta^2\sqrt{n}\log(n/\Delta^2)/\log\log(n/\Delta^2))$.

7 Concluding Remarks

A comprehensive layout strategy based on graph separators was developed for finding area-efficient layouts of graphs of arbitrary degree. The methodology was shown to be applicable to an alternative graph dissection method [7] as well. From these results, it is clear that for most characterisations of edge partition trees, the resulting layout area is independent of the maximum vertex degree. In all these cases, the bounds on the layout area obtained here are better by a factor of $O(\Delta^2)$ than corresponding bounds obtained by extending existing techniques to handle vertices of high degree. It was also shown that the layouts can be modified to minimise the maximum wire length while limiting the increase in area to a constant factor.

Although planar graphs are also $\sqrt{n}$-separable, they can be shown to have layouts which are significantly smaller than those for arbitrary $\sqrt{n}$-separable graphs. Chapter 4 of [13] presents layout schemes for planar graphs that are simultaneously sensitive to: (1) the degree of 'outerplanarity' of the graph (*gauge*), (2) the degree of 'planarity' of the layout (*contact cuts*), and (3) the maximum degree of the graph.

Considering that $o(n)$ separable graphs have only $O(n)$ edges, it is unlikely that they have many vertices of the maximum degree Δ, when Δ is large. Results have been obtained for these cases as well [13]. To this end, a characterisation has been developed for the distribution of edges over vertices under which different classes of vertex-separable graphs of any maximum degree would have the same layout area as corresponding graphs of bounded degree.

The techniques developed here also have other applications. It has been demonstrated (Chapter 2 of [13]) that the separators satisfying the *external edge balancing condition* are also useful for solving the problem of task allocation in tree-connected multiprocessors.

Acknowledgements: The research reported here was conducted while the first author was pursuing his doctoral studies at the Department of Computer Science, University of Maryland. The first author was supported by the National Science Foundation grants NFSD-CR-83-05992 and CDR-8500108, and the Air Force Office of Scientific Research grant AFOSR-82-0303. The second author was supported by the National Science Foundation grant CDR-8500108 and NASA grant NAG5-776.

References

[1] S. N. Bhatt and F. T. Leighton. A framework for solving vlsi graph layout problems. *Journal of Computer and System Sciences*, 28:300–343, 1984.

[2] S. N. Bhatt and C. E. Leiserson. *Minimizing the Longest Edge in a VLSI Layout.* Technical Report, MIT Lab. for Computer Science, 1981.

[3] S. A. Browning. Algorithms for the tree machine. In C. Mead and L. Conway, editors, *Introduction to VLSI Systems*, Addison-Weseley, Reading, MA, 1980.

[4] J. R. Gilbert. *Graph Separator Theorems and Sparse Gaussian Elimination.* PhD thesis, Stanford University, 1980.

[5] J. R. Gilbert, D. J. Rose, and A. Edenbrandt. A separator theorem for chordal graphs. *SIAM Journal on Algebraic and Discrete Methods*, 5(3):306–313, 1984.

[6] F. Harary. *Graph Theory.* Addison-Wesley, Reading, MA, 1969.

[7] F. T. Leighton. A layout strategy for vlsi which is provably good. *Proc. 14th Annual ACM Symposium on Theory of Computing*, 85–98, 1982.

[8] F. T. Leighton. New lower bound techniques for vlsi. *Proc. 22nd Annual Symposium on Foundations of Computer Science*, 1–12, 1981.

[9] C. E. Leiserson. Area efficient graph layouts (for vlsi). *Proc. 21st Annual Symposium on Foundations of Computer Science*, 270–281, 1980.

[10] C. E. Leiserson. *Area Efficient VLSI Computation.* PhD thesis, Carnegie Mellon University, 1981.

[11] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, Aug 1980.

[12] A. Rosenberg. Encoding data structures in trees. *Journal of ACM*, 26(4):668–689, October 1979.

[13] D. D. Sherlekar. *Graph Dissection Techniques for VLSI and Algorithms.* PhD thesis, University of Maryland, College Park, Maryland, 1987.

[14] C. D. Thompson. *A Complexity Theory for VLSI.* PhD thesis, Carnegie Mellon University, 1980.

[15] J. D. Ullman. *Computational Aspects of VLSI.* Computer Science Press, Rockville, MD, 1984.

[16] L. G. Valiant. Universality considerations in vlsi circuits. *IEEE Transactions on Computers*, C-30(2):135–140, Feb 1981.

A LP APPROACH FOR OPTIMAL RECTANGLE ROUTING

Yuan Y. Sung
Department of Mathematical Sciences
Memphis State University
Memphis, TN 38152

Abstract

Given a rectangular circuit component with terminals around its boundary, the problem of connecting a set of terminals to minimize the total area is considered. We present a new approach to solve this problem by converting it into a linear programming (LP) problem. Although the solution obtained by solving the LP problem gives a lower bound value, it may not be feasible. However, we find the LP solution can be used to generate a feasible solution D. The difference between D and an optimal solution is no more than one track per direction (horizontal or vertical).

1. INTRODUCTION

Wire routing around a rectangle is the problem of connecting terminals located along the boundary of a circuit component in such a way that the total area is minimized. The problem has been defined by [4] and [1] as below.

Given a rectangle and a set of terminals that lie on the sides of the rectangle. Terminals to be interconnected are called nets. Wires of uniform width are used to connect the terminals in each net. The path followed by these wires consist of a finite number of horizontal and vertical line segments. These line segments can be assigned to two different layers. One layer consists of all the horizontal line segments and the other layer consists of all the vertical ones. A wire may change layers at any point. Wires on different layers may cross. A minimum spacing is required between any two parallel line segments. No path is allowed inside the rectangle on any of the layers. We want to arrange paths for all the connecting wires such that the total area is minimized.

The particular problem we consider in this paper has only one component, and each net contains two terminals. This problem is interesting because it is one of a few routing problems which are believed to be solvable in polynomial time. This problem has applications in the layout of integrated circuits ([4], [6]) and conforms to a set of design rules for VLSI systems [5].

For simplicity, we call the above problem RR (routing around rectangle) problem and use this term throughout the paper. LaPaugh presented an $0(n^3)$ algorithm to solve the RR problem where n is the number of terminals. Gonzalez and Lee solved the RR problem with an $0(nlogn)$ algorithm. In this paper we present a new approach to solve this problem. Our algorithm uses linear programming (LP) to firstly derive a lower bound value L, then generate a feasible solution which is larger than L by at most one on each direction (vertical or horizontal). The difference between the approach presented in this paper and previous papers is that our concept is much easier to understand; also the proof is very simple. Our approach has two stages, the first stage is modeling and solving a LP problem, the second one is actually deriving a feasible solution. Therefore, the first stage of our approach can be used as a quick near-optimal estimation for the RR problem; that is very useful because many times in the layout design process we probably only need a good and quick estimation.

In section 2, some of the previous results which we need in this paper are described. In section 3, we present and explain the LP model that we will use to solve the RR problem. We also show the derivation of a near optimal feasible solution from the LP result. The complexity analysis of our algorithm is presented in section 4, and some improvements in computational efficiency are discussed in details. Section 5 gives a brief conclusion and some future research topics.

2. SOME PREVIOUS WORKS

LaPaugh [4] has outlined an algorithm that solves the RR problem in $0(n^3)$ time, where n is the number of terminals on the component. Some of LaPaugh's major contributions are: (1) All of the local terminals should be connected by paths crossing at most one corner of the component. (Pairs of terminals on the same side or adjacent side of the component are called local terminals; otherwise, they are global terminals); (2) The choice of directions for top-bottom connections is independent of the direction of left-right connections and vice-versa; (3) For every assignment, a final layout can be realized in $0(nlogn)$ time. Therefore, LaPaugh has shown the RR problem can be reduced into two instances of the following problem:

Given all global terminals appear on the top and bottom sides, decide which direction each connection should go so that the resulting total vertical dimension (height) is minimized.

From now on we shall assume that the RR problem we will study is the above restricted version. That is, all global terminals are located on the top and bottom sides, and each local connection is made by going the shortest way.

Gonzalez and Lee [1] present an $0(nlogn)$ algorithm to solve this problem. Their strategy is to reduce the problem to several subproblems that have the property that at least one of these subproblems can be solved optimally by the greedy method. One of the most important results in [Gon82] is that the existence of an optimal solution D which satisfies the following properties:

(a) there are two points α and β on the top side of the rectangle such that all global terminals located on the left of α are connected by a path crossing the left side of the rectangle, and all global terminals located on the right of β are connected by a path crossing the right side of the rectangle;

(b) $|h_\alpha(D) - h_\beta(D)| \leq 1$;

(c) $h_{\text{top}}(D) - h_\alpha(D) \leq 1$ and $h_{\text{top}}(D) - h_\beta(D) \leq 1$,

where $h_x(D)$ is the height at a point x by the solution D and $h_{\text{top}}(D)$ is the height of the top by the solution D. That is, $h_x(D)$ = the number of paths given by D that cross point x, and $h_{\text{top}}(D) = max\{h_x(D)|\ x$ is a point on the top $\}$.

In our approach, we use properties that are similar to the above (a) - (c) to improve the computational efficiency. The details can be found in section 4.

3. OPTIMAL RECTANGLE ROUTING BY LINEAR PROGRAMMING

In this section, we first show that the RR problem can be exactly represented by an integer programming (IP) model. For computational tractability, we then extend the IP model into a linear programming (LP) model. Solving the LP model gives a lower bound. In the second stage, we show that the linear solution obtained can be converted into a feasible integer solution, and their cost (or height) difference is at most one.

Suppose we have n global terminals $T_1, T_2, \ldots, T_n$ on the top, and their corresponding terminals on the bottom are $T_{C(1)}, T_{C(2)}, \ldots, T_{C(n)}$. Let t_i be the location of terminal T_i. Without loss of generality, we can assume that $t_1 < t_2 < \ldots < t_n$, that is, $T_1, T_2, \ldots, T_n$ are resided in order of left to right; furthermore, we assume t_0 as the left boundary and t_{n+1} as the right boundary.

We define n integer variables $x_1, x_2, \ldots, x_n$, each has value 0 or 1, as follows:

$x_i = 1$ if and only if the connection between T_i and $T_{C(i)}$ is by passing the left side of the rectangle.

An assignment D for the RR problem now becomes an assignment of 0 or 1 to the n variables $x_1, x_2, \ldots, x_n$.

Let $X = (x_1, x_2, \ldots, x_n)$ be an assignment for the RR problem. Let $\overline{h}_a$ and $h_a(X)$ be the heights at terminal a before and after the assignment of X, respectively. (Note that $\overline{h}_a$ is fixed and does not depend on X). For $i = 0, 1, \ldots, n$, we define the region i, denoted by R_i, as the interval between t_i and t_{i+1}. (i.e., $R_i = [t_i, t_{i+1}]$).

For a terminal a in region R_0, we have $h_a(X) = \overline{h}_a + x_1 + x_2 + \ldots + x_n$. Since we are only interested in finding the maximal height, it is therefore only necessary to keep the terminal a_1 which has maximum $\overline{h}$ value among all terminals in R_0; and all other terminals in R_0 can be discarded from further consideration. We denote the maximum height in region R_1 as $y_1(X)$, or simply y_1 if X is understandable. Similarly, for $i = 1, \ldots, n$, we can find the maximum height in region R_i is $y_i = -x_1 - x_2 - \ldots - x_i + x_{i+1} + \ldots x_n + c_i$, where $c_i = i + \overline{h}_{a_i}$ is a constant and a_i is a terminal at which R_i has its maximum $\overline{h}$ value. Analogously, we can obtain the maximum heights for all regions of the bottom side.

3.1 A LP Model for the RR Problem

The above discussions show that finding an optimum solution for the RR problem is the same as solving the following integer programming problem IP1:

(IP1): $\min\ (\max\{y_0, \ldots, y_n\} + \max\ \{z_0, \ldots, z_n\})$

s.t.

$$w = A \bullet x + c$$

where $$\begin{aligned} w^T &= [y_0 \dots y_n \; z_0 \dots z_n], \\ x^T &= [x_1 \dots x_n], \quad \text{with} \quad x_i = 0 \quad \text{or} \quad 1 \quad \text{for} \quad 1 \le i \le n, \\ c^T &= [c_0 \dots c_n \; c'_0 \dots c'_n], \quad c_i \text{ and } c'_i \text{ are positive integers,} \\ \text{and } A &= [a_{ij}], \quad a_{ij} = -1 \quad \text{or} \quad 1, \quad \text{for all} \quad i,j, \quad 1 \le i \le 2n+2, \quad 1 \le j \le n. \end{aligned}$$

The matrix A and vector c are predetermined and considered as constants. The vectors w and x are variables.

Let $y = max\{y_0, \dots, y_n\}$ and $z = max\{z_0, \dots, z_n\}$. The above problem IP1 can be easily verified to be equal to the following IP2 problem:

(IP2): $\min_x \{y + z\}$

s.t.

$$w \ge A \bullet x + c$$

where

$$w^T = [y \dots y \; z \dots z] \text{ is a vector of } (n+1) \; y \text{ and } (n+1) \; z,$$

$$\text{and } \; A, x, c \; \text{ are defined the same as IP1.}$$

Although the RR problem can be nicely represented by an IP model, it may not be computationally solvable in reasonable time since the IP problem is intractable in general. However, we can extend the IP2 model into a linear programming (LP) model that can be solved efficiently.

(LP): $\min_x \{y + z\}$

s.t.

$$w \ge A \bullet x + c$$

where

$$w, A, c \; \text{ are defined as IP2, and } \; x^T = [x_1, \dots, x_n], \text{with } 0 \le x_i \le 1, \text{ for } 1 \le i \le n.$$

Clearly, the minimum value obtained by solving LP1 is a lower bound for the RR problem; however, the solution x may not be feasible.

3.2 Conversion a LP Solution to a Feasible Solution

Suppose the solution $x = (x_1, \dots, x_n)$ is not feasible because some of the values are not integer. Without loss of generality, we can assume the first m variables $x_1, \dots, x_m$ are infeasible. That is, $0 < x_i < 1$ for $1 \le i \le m$.

For a pair of nets $N_i = (T_i, T_{C(i)})$ and $N_j = (T_j, T_{C(j)})$ with $1 \le i < j \le m$, there are four kinds of transformation.

Type-1 transformation: $x_i \leftarrow x_i + \Delta \; ; \; x_j \leftarrow x_j - \Delta,$

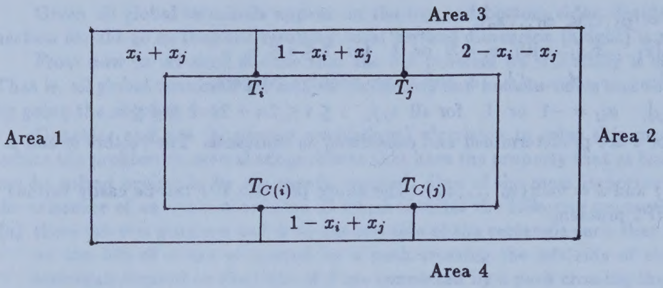
Type-2 transformation: $x_i \leftarrow x_i - \Delta; \; x_j \leftarrow x_j + \Delta,$

Type-3 transformation: $x_i \leftarrow x_i - \Delta; \; x_j \leftarrow x_j - \Delta,$

Type-4 transformation: $x_i \leftarrow x_i + \Delta; \; x_j \leftarrow x_j + \Delta.$

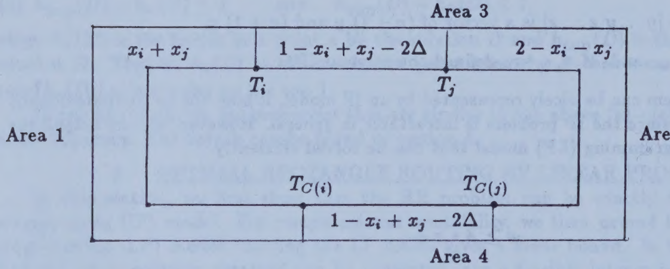
The amount Δ in each case should be selected such that after the transformation is done, the new x_i and x_j still preserve the property that they remain in the range of $[0, 1]$.

For the pair of nets N_i and N_j, there are two cases, either $T_{C(i)} < T_{C(j)}$ or $T_{C(i)} > T_{C(j)}$. In the former case, as shown in figure 1(a), the amount of x_i and x_j involved in the height $h_a(X)$ for a point a in area-1 is $x_i + x_j$. For areas 2, 3 and 4, their values are $(1 - x_i) + (1 - x_j)$, $(1 - x_i) + x_j$ and $(1 - x_i) + x_j$, respectively. By using type-1 transformation and selecting min $\{1 - x_i, x_j\}$ as Δ, the new values are shown as in figure 1(b). The change of height before and after the transformation is shown in figure 1(c). The heights in area 1 and 2 remain unchanged, but decreased by 2Δ in area 3 and 4. This shows that the minimum value (or global height) obtained by LP solution is unchanged by this transformation. It is clear that such a choice of Δ preserves the property that the new x_i and x_j are still in the range of $[0, 1]$. In fact, either x_i increases to 1 (if $\Delta = 1 - x_i$) or x_j decreases to 0 (if $\Delta = x_j$) or both.



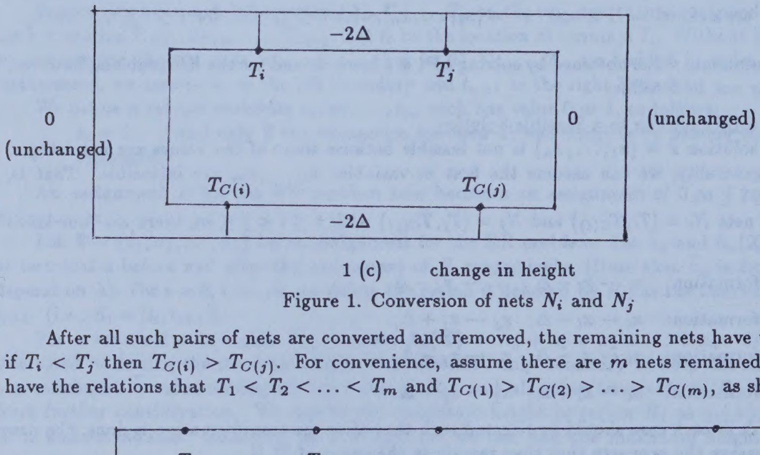
Area 3
$x_i + x_j$
$1 - x_i + x_j$
$2 - x_i - x_j$
T_i
T_j
Area 1
Area 2
$T_{C(i)}$
$T_{C(j)}$
$1 - x_i + x_j$
Area 4

1 (a) before transformation



Area 3
$x_i + x_j$
$1 - x_i + x_j - 2\Delta$
$2 - x_i - x_j$
T_i
T_j
Area 1
Area 2
$T_{C(i)}$
$T_{C(j)}$
$1 - x_i + x_j - 2\Delta$
Area 4

1 (b) after transformation

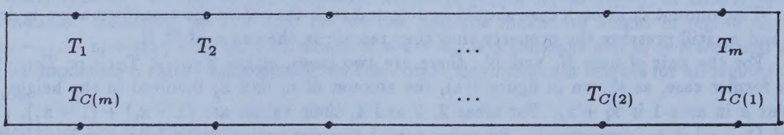


-2Δ
T_i
T_j
0
(unchanged)
0
(unchanged)
$T_{C(i)}$
$T_{C(j)}$
-2Δ

1 (c) change in height

Figure 1. Conversion of nets N_i and N_j

After all such pairs of nets are converted and removed, the remaining nets have the property that if $T_i < T_j$ then $T_{C(i)} > T_{C(j)}$. For convenience, assume there are m nets remained, then they must have the relations that $T_1 < T_2 < \ldots < T_m$ and $T_{C(1)} > T_{C(2)} > \ldots > T_{C(m)}$, as shown in figure 2.



T_1
T_2
...
T_m
$T_{C(m)}$
...
$T_{C(2)}$
$T_{C(1)}$

Figure 2. after conversion step 1

Now every pair of nets $N_i = (T_i, T_{C(i)})$ and $N_j = (T_j, T_{C(j)})$ have $T_i < T_j$ and $T_{C(i)} > T_{C(j)}$, as shown in figure 3. Applying type-1 transformation does not change values in area 1 and 2; it reduces an amount of 2Δ in area 3 but increases 2Δ in area 4. On the other hand, type-2 transformation makes area 4 loses 2Δ but area 3 gains 2Δ while area 1 and 2 remain the same. Of course, to preserve that the new

x_i and x_j remain in $[0,1]$ and to make one or two of them to be integer, we select $\Delta = min\{1-x_i, x_j\}$ if type-1 transformation is employed and $\Delta = min\{x_i, 1-x_j\}$ if type-2 transformation is used.

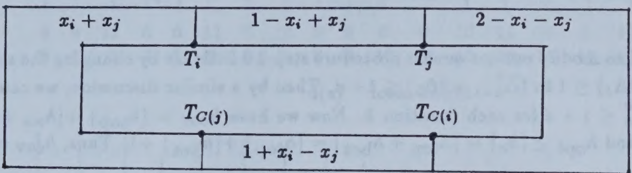


Figure 3. Two nets N_i and N_j

The second step of conversion is that always selecting the leftmost (unconverted) net as N_i and the rightmost one as N_j (i.e., N_i moves from left to right and N_j moves from right to left). At each iteration step, the height change in the area $[T_i, T_j]$ and $[T_{C(j)}, T_{C(i)}]$ depends on the previous accumulated height change and the current Δ value (of either type-1 or type-2 transformation).

Let the accumulated height change on the top side at k^{th} iteration be $\overrightarrow{\Delta}^{t}_{k}$ and the accumulated height change on the bottom side be $\overrightarrow{\Delta}^{b}_{k}$. It is clear that if type-1 transformation is used, then $\overrightarrow{\Delta}^{t}_{k} = \overrightarrow{\Delta}^{t}_{k-1} - 2\Delta_1$ and $\overrightarrow{\Delta}^{b}_{k} = \overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1$ where $\Delta_1 = min(1-x_i, x_j)$. On the other hand, if type-2 transformation is chosen, then $\overrightarrow{\Delta}^{t}_{k} = \overrightarrow{\Delta}^{t}_{k-1} + 2\Delta_2$ and $\overrightarrow{\Delta}^{b}_{k} = \overrightarrow{\Delta}^{b}_{k-1} - 2\Delta_2$ where $\Delta_2 = min(x_i, 1-x_j)$. It is desired that if the accumulated height change at each iteration is always not to exceed 1 on both top and bottom sides. This goal can be achieved if we select the transformation as follows: Type-1 transformation is used if $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) \leq 1$ where $\Delta_1 = min\ (1-x_i, x_j)$. Otherwise, type-2 transformation is chosen.

Lemma 3.1 (a) $\overrightarrow{\Delta}^{t}_{k} = -\overrightarrow{\Delta}^{b}_{k}$.

(b) $\overrightarrow{\Delta}^{t}_{k} < 1$ and $\overrightarrow{\Delta}^{b}_{k} \leq 1$ for all k.

Proof: By induction on k. Define $\overrightarrow{\Delta}^{t}_{0} = \overrightarrow{\Delta}^{b}_{0} = 0$.

Note that $\Delta_1 + \Delta_2 \leq 1$ is always true.

(i) $k = 1$. If $2\Delta_1 \leq 1$ then it is obviously true. Otherwise, type-2 transformation is chosen, $\overrightarrow{\Delta}^{t}_{1} = 2\Delta_2$ and $\overrightarrow{\Delta}^{b}_{1} = -2\Delta_2$. Since $2\Delta_1 > 1$ and $\Delta_1 + \Delta_2 \leq 1$, we get $2\Delta_2 < 1$. Thus, it is also true.

(ii) Assume $\overrightarrow{\Delta}^{t}_{k-1} < 1$ and $\overrightarrow{\Delta}^{b}_{k-1} \leq 1$, and $\overrightarrow{\Delta}^{t}_{k-1} = -\overrightarrow{\Delta}^{b}_{k-1}$. If $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) \leq 1$, then $\overrightarrow{\Delta}^{b}_{k} = \overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1$ and $\overrightarrow{\Delta}^{t}_{k} = \overrightarrow{\Delta}^{t}_{k-1} - 2\Delta_1$, it is obviously true. If $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) > 1$, then $\overrightarrow{\Delta}^{t}_{k} = \overrightarrow{\Delta}^{t}_{k-1} + 2\Delta_2$ and $\overrightarrow{\Delta}^{b}_{k} = \overrightarrow{\Delta}^{b}_{k-1} - 2\Delta_2$. It is clear that $\overrightarrow{\Delta}^{t}_{k} = -\overrightarrow{\Delta}^{b}_{k}$. Since $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) > 1$ and $\Delta_1 + \Delta_2 \leq 1$, we get $\overrightarrow{\Delta}^{b}_{k-1} - 2\Delta_2 > -1$. Therefore, we obtain $\overrightarrow{\Delta}^{b}_{k} > -1$ and so $\overrightarrow{\Delta}^{t}_{k} = -\overrightarrow{\Delta}^{b}_{k} < 1$. Q.E.D.

The above lemma 3.1 states that the maximum height change on each side (top or bottom) is at most one. In particular, it is strictly less than 1 in one of the two sides.

Theorem 3.1 The maximum height difference between the optimal solution and the solution are obtained after the conversion process is at most one per direction (vertical or horizontal).

Proof. Let h_x, h_{opt} and h_{cov} be the global (maximum) heights of the linear solution, the optimal solution and the feasible solution obtained after the conversion, respectively. From lemma 3.1, we have $h_{\text{cov}} < h_x + 2$ and h_{cov} is an integer. Since $h_x \leq h_{\text{opt}}$, we have $0 \leq h_{\text{cov}} - h_{\text{opt}} \leq h_{\text{cov}} - h_x < 2$. Because h_{cov} and h_{opt} both are integers, we get $h_{\text{cov}} - h_{\text{opt}} = 0$ or 1. Q.E.D.

Corollary 3.1 The solution obtained by the conversion procedure is optimal if the LP solution has a height that is not an integer.

Proof. Let $h_x = h_{\text{top}} + h_{\text{bot}}$ be the global height obtained by the LP solution where h_{top} and h_{bot} are the global heights of the top side and bottom side, respectively. Let h_{opt} and h_{cov} be the heights of the optimal solution and the feasible solution by the conversion procedure, respectively.

Let $d = h_{\text{top}} - \lfloor h_{\text{top}} \rfloor$ and $d' = h_{\text{bot}} - \lfloor h_{\text{bot}} \rfloor$. Since we assume h_x is not an integer, it is not hard to see $0 < d + d' < 2$ and $d + d' \neq 1$.

Case 1. $1 < d + d' < 2$.

In this case, $h_{\text{opt}} \geq \lceil h_x \rceil = \lceil h_{\text{top}} + h_{\text{bot}} \rceil = \lfloor h_{\text{top}} \rfloor + \lfloor h_{\text{bot}} \rfloor + 2$. By lemma 3.1, we know $h_{\text{cov}} \leq \lfloor h_{\text{top}} + 1 \rfloor + \lfloor h_{\text{bot}} + 1 \rfloor = \lfloor h_{\text{op}} \rfloor + \lfloor h_{\text{bot}} \rfloor + 2$. Therefore, $h_{\text{cov}} = h_{\text{opt}}$ and the solution is an optimal solution.

Case 2. $0 < d + d' < 1$.

In this case, we need to modify our conversion procedure step 2 a little bit by changing the selection criterion from $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) \leq 1$ to $(\overrightarrow{\Delta}^{b}_{k-1} + 2\Delta_1) \leq 1 + d$. Then by a similar discussion, we can derive that $\overrightarrow{\Delta}^{t}_{k} < 1 - d$ and $\overrightarrow{\Delta}^{b}_{k} \leq 1 + d$ for each iteration k. Now we have $h_{\text{cov}} = \lfloor h_{\text{top}} \rfloor + \lfloor h_{bot} + 1 + d \rfloor = \lfloor h_{\text{top}} \rfloor + \lfloor h_{\text{bot}} \rfloor + 1$, and $h_{\text{opt}} \geq \lceil h_x \rceil = \lceil h_{\text{top}} + h_{\text{bot}} \rceil = \lfloor h_{\text{top}} \rfloor + \lfloor h_{\text{bot}} \rfloor + 1$. Thus, $h_{\text{cov}} = h_{\text{opt}}$. Q.E.D.

In the above discussions we assume that after the termination of algorithm conversion, all of the variables are converted. It is not hard to see that this assumption is true if and only if the global height at region 0 is an integer. (Note that the global height at region 0 is the same before and after the algorithm conversion). Let h_0 be the global height (of the assignment X) at region 0. In the case of h_0 is not an integer, before the start of algorithm conversion, if we firstly increase h_0 to $\lceil h_0 \rceil$ then we can make certain that all variables are to be converted by algorithm conversion. The increase of global height at region 0 from h_0 to $\lceil h_0 \rceil$ can be done by applying a sequence of type-4 transformations. For each type-4 transformation, the pair of nets $N_i = (T_i, T_{C(i)})$ and $N_j = (T_j, T_{C(j)})$ selected are always that N_i is the leftmost net on the top and N_j is the leftmost net on the bottom. The interested reader can easily verify that the above theorems still hold.

4. ANALYSIS OF COMPUTATIONAL COMPLEXITY

Considering the computational time, the most expensive part in our approach is the step of solving the LP model. Other steps can be solved in $0(n \log n)$ time.

Two important factors that affect the computational time in solving a LP problem are:

(1) the size of the LP model, and

(2) a good initial solution.

In this section, we discuss some methods that reduce the size of the LP model, and some that give a good initial point.

4.1 To reduce the size of LP model

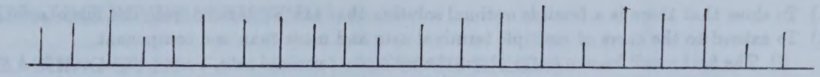
Each row in our LP model corresponds to a region. Let X be the assignment of assigning all $x_i = 1$, $i = 1, \ldots, n$. That is, all the terminals on the top are to be wired to the left. For any region, i, $o \leq i \leq n$, if there exists a region j with $j > i$ and $h_X(j) \geq h_X(i)$, then it is not hard to see that the height of region i can never exceed the height of region j under any assignment. This implies that region i does not contribute anything to the maximum height we are looking for, and consequently its corresponding row can be removed from the LP model. To find out all such unuseful regions described as above, a scanning from right to left can be carried out. In the process of scanning, if region i is currently under examination, then its height $h_X(i)$ is compared with the maximum height h obtained so far; if $h_X(i) \leq h$, then region i is eliminated from further consideration; otherwise, h is updated by $h_X(i)$. Similarly, an assignment $\overline{X}$ that sets value 0 to all variables x_i, $i = 1, \ldots, m$, can be used to detect and then remove a region that does not contribute (i.e., it has a height less than or equal to the height of some region on its left).

The unnecessary regions on the bottom side can also be deleted in an analogous manner.

Suppose region i is the leftmost region that is not removed. It is clear that, for $k = 1, \ldots, i-1$, if we assign 0 to a variable x_k then the maximum height on the top side is definitely increased by one. That is, there always exists an optimal assignment by assigning 1 to all variables x_k, $1 \leq k \leq i-1$. Similarly, all variables x_k, $j+1 \leq k \leq n$, can be assigned with a value 0, where j is the rightmost region not eliminated. (Note: i and j here are the same as α and β defined in [1].)

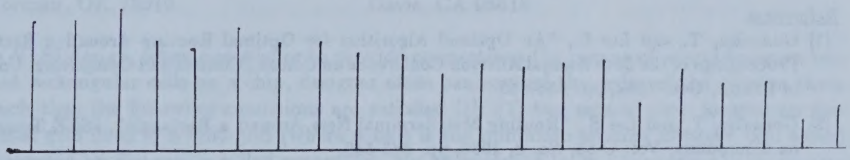
Example 4.1 Suppose there are 20 global terminals on the top side, and the local heights of the 21 regions are 8, 9, 12, 6, 6, 11, 9, 10, 9, 9, 6, 9, 10, 11, 13, 5, 13, 11, 12, 5, 7, as shown in figure 4(a).

After assigning all $x_i = 1$ for $i = 1, \ldots, 20$, the heights are shown in figure 4(b). A scan from right to left eliminates all the regions that do not contribute to the LP model. The regions still left not eliminated are those circled by squares. Similarly, for an assignment of $x_1 = \ldots = x_{20} = 0$, the heights become the ones shown in figure 4(c). After removing the unuseful regions 8 and 20, there are only seven regions left now. The leftmost region is 2 and the rightmost region is 18. Therefore, there exists an optimal solution with $x_0 = x_1 = 1$ and $x_{19} = x_{20} = 0$.



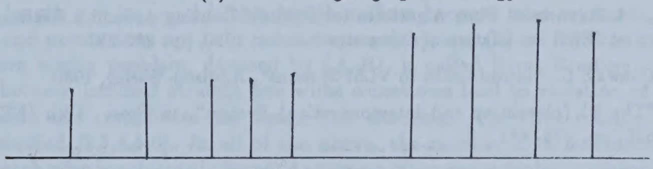
height:	8	9	12	6	6	11	9	10	9	9	6	9	10	11	13	5	13	11	12	5	7
region:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

4(a) local height



height:	28	28	30	23	22	26	23	23	21	20	16	18	18	18	19	10	17	14	14	6	7
region:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

4(b) after assigning $x_1 = \ldots = x_{20} = 1$



height:	14	16	17	17	18	27	29	30	27
region:	2	5	7	8	9	14	16	18	20

4(c) after assigning $x_1 = \ldots = x_{20} = 0$

Figure 4. Example 4.1

4.2 To find a good initial point

To solve a LP problem, by and large, the closer an initial solution to an optimal solution, the quicker result can be obtained.

Suppose now all the non-contributory regions have been removed. Assume there are n global terminals need to be assigned, α and β are the leftmost and rightmost terminals, respectively. Let the (local) height of region $(\alpha - 1)$, before any assignment, be a, and the (local) height of region $(\beta + 1)$ be b. It is clear that the maximum (global) height on the top is at least as much as $\lceil (a + b + n)/2 \rceil$. Suppose the n global variables need to be assigned are $x_1, \ldots, x_n$. A good initial solution of our LP model is the following:

Case 1. $(a + b + n)$ is even.

Let $p = (a + b + n)/2 - a$. Set $x_1 = \ldots = x_p = 1$, and $x_{p+1} = \ldots = x_n = 0$.

Case 2. $(a + b + n)$ is odd.

Let $p = \lfloor (a + b + n)/2 \rfloor - a$. Set $x_1 = \ldots = x_p = 1$, $x_{p+2} = \ldots = x_n = 0$, and $x_{p+1} = 0.5$

Example 4.2 In the example 4.1, $n = 16$, $\alpha = 2$ and $\beta = 18$. Thus, $a = 12$ and $b = 12$. Therefore, a good initial point is to assign $x_1 = x_2 = \ldots = x_8 = 1$ and $x_9 = \ldots x_{16} = 0$.

5. CONCLUSIONS AND FUTURE RESEARCHES:

In this paper a new approach for solving the routing problem around a rectangular component is described. The new approach employs a LP model. It is interesting to know that the linear solution only differs from the optimal 0/1 integer solution by at most one in each direction. Therefore, our method gives a quick estimation, and in many layout design problems that is a desirable feature.

Some of the future research topics are:

(1) To show that there is a feasible optimal solution that can be derived from the linear solution.

(2) To extend to the cases of multiple terminal nets and more than one component,
  (i) The best result known so far about the multiple- terminal nets, 1 component case is $A \leq 1.6A^*$, where A^* is the optimal solution. ([2])
  (ii) The best approximation result known about two-terminal, 2-components case is $A \leq 2A^*$, where A^* is the optimal answer. ([3])

References

[1] Gonzalez, T. and Lee S., "An Optimal Algorithm for Optimal Routing Around a Rectangle", Proceedings of the 20th Annual Allerton Conference on Comm., Control and Computing, University of Illinois, Oct. 1982, pp. 636-645.

[2] Gonzalez, T. and Lee S., "Routing Multiterminal Nets Around a Rectangle", IEEE Transactions on Computers, Vol. C-35, No. 6, 1986, pp. 543-549.

[3] Gonzalez, T. and Lee S., "An Approximation Algorithm for Routing Two-Terminal Nets Around Two Rectangles", Proceedings of the 24th Annual Allerton Conference on Comm., Control and Computing, University of Illinois, Oct. 1986, pp. 550-559.

[4] LaPaugh, A.S., "A Polynomial Time Algorithm for Optimal Routing Around a Rectangle", Proceedings of the 21st IEEE Foundation of Computer Science, 1980, pp. 282-293.

[5] Mead, C. and Conway, L., "Introduction to VLSI Systems", Addison-Wesley, 1980.

[6] Rivest, R.L., "The PI (placement and interconnection) System", in Proc. 19th IEEE Design Automation Conf., pp. 475- 481.

SINGLE JOG RIVER ROUTING[1]

TAI-CHING TUAN
School of Electrical Engineering
and Computer Science
University of Oklahoma
Norman, OK 73019

S. LOUIS HAKIMI
Department of Electrical
and Computer Engineering
University of California
Davis, CA 95616

In VLSI design, when one wishes to provide connections (by means of wires) between two adjacent rectangular cells on a chip, designer often can control the order of the pins on these cells such that the following conditions are satisfied [1]: (1) two sets of pins, located on two horizontal grid lines in a unit-grid (where a unit is the minimum spacing between two wires) and separated by distance s called separation, are represented by two sequences of increasing integers $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$ where A is the x-coordinates of the pins on one cell and B is those on the other; (2) a wire representing the net N_i, for $i = 1, 2, \ldots, n$, must join the terminal at a_i to the terminal at b_i by means of a continuous rectilinear curve of total length $s + |a_i - b_i|$ (this length suffices because wire segments extending beyond the two end points does not help reduce separation); and (3) no wire can touch another wire. The above wiring problem, denoted by (A, B), is called River Routing. Rectilinear wiring is used because inclined straight line wires sometimes lead to violation of minimum spacing requirement and complicate the design rule checking. The river routing problem has been widely studied [2,3,4,5,6]. In all of the above, the number J of horizontal segments (called jogs) in each wire is arbitrarily large. As corners often cause faults, a wiring in which each wire has fewer jogs is more desirable particularly when it does not (substantially) increase s. Let w be an integer, called offset, and $B(w) = (b_1 + w, b_2 + w, \ldots, b_n + w)$. The integer pair (s, w) is said to be feasible, if there is a feasible wiring of $(A, B(w))$ with separation s. Let $s^*(w)$ be the smallest separation for which $(s^*(w), w)$ is feasible. However, it appears that any of the problems studied in [2,3,4,5,6] becomes very difficult when an upper bound on J is placed. When $J = 2$ the river routing problem becomes much more difficult. We have developed a dynamic programming algorithm for finding $s^*(w)$ which will be reported in another paper. For the remaining of this paper, we assume that $J = 1$.

Lemma 1: Let T_1 (respectively, T_2) be the largest set of consecutive integers $[i, i + k]$, in the interval $[1, n]$ such that $a_{j+1} - b_j \leq w$ for all j in T_1 (respectively, $a_j - b_{j+1} \geq w$ for all j in T_2). Then for a wiring of $(A, B(w))$ in which $J = 1$, $s^*(w) = \max\{|T_1|, |T_2|\} + 2$.

The above lemma is used to design a simple linear time algorithm for computing the smallest separation $s^*(w)$ given $(A, B(w))$. Let $\ell_j = a_j - b_{j+1} + 1$, and $u_j = a_{j+1} - b_j - 1$. Define $L'(s)$ and $U'(s)$ as follows. $L'(1) = \max\{a_j - b_j | 1 \leq j \leq n\}$ and $U'(1) = \min\{a_j - b_j | 1 \leq j \leq n\}$. For $s > 1, L'(s) = \max\{\min\{\ell_j, \ell_{j+1}, \ldots, \ell_{j+s-2}\} | 1 \leq j \leq n - s + 1\}$ and $U'(s) = \min\{\max\{u_j, u_{j+1}, \ldots, u_{j+s-2}\} | 1 \leq j \leq n - s + 1\}$. It is easy to see that $(U'(s) - L'(s))$ is an increasing function of s. Define $(W, S) = \{(s, w) | (s, w) \text{ is feasible }\}$. For given s, let $W(s) = \{w | (s, w) \text{ is a feasible pair}\}$. Let w^* be such that $s^*(w^*) \leq s^*(w)$ for all w.

Theorem 1: (s, w) is feasible if and only if w is in $W(s) = [L'(s), U'(s)]$.

If $s = 1$, then $L'(1)$ and $U'(1)$ can be computed in linear time. To compute $L'(s)$ for $s > 1$, an $(s - 1)$-node complete binary tree structure is used. Each node of the structure contains two components - one is the value ℓ_i itself, the other is the index i of ℓ_i. A complete binary tree is called a partial heap if the key of the root is the extreme. Initially a partial heap is set up for the sequence from ℓ_1 to ℓ_{s-1} in a bottom up manner. It takes $0(\log(s - 1))$ comparisons to form a partial heap. Output the minimum element which is contained in the first node of the partial heap. Then do the following: 1) identify the node for the first element ℓ_i of the sequence $(\ell_i, \ell_{i+1}, \ldots, \ell_{i+s-2})$ and change its two components to ℓ_{i+s-1} and $i+s-1$, 2) maintain the partial heap property and 3) output the first component of the first node. After this, the tree structure is for the sequence from ℓ_{i+1} to ℓ_{i+s-1}. In order to directly locate the node for ℓ_i in the structure, an array $T[1..n-1]$ with its $i-th$ entry, $T(i)$, being the location of the node containing ℓ_i in the structure is maintained. Each time when the node for ℓ_i and the node for ℓ_j are exchanged in the structure, the contents of $T(i)$ and $T(j)$ are also exchanged. The above process will continue until no new $\ell_i's$ can enter into the structure. Finally, the maximum of the $(n-s+1)$ output minima is computed and assigned to $L'(s)$. $U'(s)$ for $s > 1$ is computed similarly. Based on Theorem 1 and the above discussion, $0((n-s)\log\,(s-1)+n)$ time is needed to compute $W(s)$ for each s.

---

[1] This work is supported by NSF Grants DMC–8406854 and ECS–8511211

Suppose momentarily that $s > 1$ is fixed. Let array $AL_s[1..n-s+1]$ be such that its j-th entry, $AL_s(j)$, is the minimum in the sequence from ℓ_j to ℓ_{j+s-2} and array $AU_s[1..n-s+1]$ be such that its j-th entry, $AU_s(j)$, is the maximum in the sequence from u_j to u_{j+s-2}. Then $L'(s)$ is the maximum in AL_s and $U'(s)$ the minimum in AU_s. (W,S) can be computed as follows. $W(1)$ is obtained by computing $L'(1)$ and $U'(1)$. If $n=1$, then stop; otherwise compute $AL_2[1..n-1]$ and determine $L'(2)$. When $j > 2$ and $AL_{j-1}[1..n-j+2]$ is given, $AL_j(i) = \min\{AL_{j-1}(i), \ell_{i+j-2}\}$ for $1 \le i \le n-j+1$ and $L'(j)$ is determined thereafter. For $j > 1$, $U'(j)$ can be computed similarly. In total, the time complexity for computing (W,S) is $0(n^2)$. The above idea can also be used for (1) finding an offset w^* which achieves the least area required to solve $(A, B(w))$ in $0(n^2)$ time; and (2) finding a feasible pair (s,w) which achieves the shortest total wire length, $ns + \Sigma_{i=1}^{n}|a_i - b_i - w|$, in $0(n^2)$ time.

The optimum offset problem is to find the smallest separation s^* which makes $W(s)$ nonempty. Two-phase binary search technique can be applied to determine s^*. There are at most $(1 + 2\,\lceil\log_2[s^*-1]\rceil)$ values of s to be examined. For each s, it takes $0(n\ \log\ s)$ to compute $U'(s)$ and $L'(s)$. As $s^* \le n$, in all, it takes $0(n\ \log^2 n)$ time with $0(n)$ space for holding $\ell_j's$ and $u_j's$. If AL_s and AU_s are also provided for each s being examined, then the time complexity can be improved to $0(n\ \log\ n)$ at the expense of $0(n\ \log\ n)$ space.

An offset w^* achieving a layout with minimum total jogs will straighten the largest number of nets. To find w^*, let $d_j = a_j - b_j$ and define $\mathrm{PLACES}(w) = \{j | w = d_j\}$. We start with sorting $d_j's$ by any efficient sorting technique and create the sets PLACES(w)'s accordingly. This, along with the determination of w^* which maximizes $|\mathrm{PLACES}(w)|$, can be done in at most $0(n\ \log\ n)$ steps. If open hash table technique [8] is used instead, PLACES(w)'s can be created in linear time on the average and w^* is determined in $0(n)$ time. Therefore only $0(n)$ time is needed on the average to find w^*.

REFERENCES

[1] Ullman, J. D., "Computational Aspects of VLSI", Computer Science Press, MD, 1984.

[2] Dolev, D.; Karplus, K.; Siegel A.; Strong, A.; Ullman, J.D.; "Optimal Wiring between Rectangles", Proc. 13th ACM SOTC, 1981, pp. 312-317.

[3] Siegel, A.; Dolev, D.; "The Separation for General Single-Layer Wiring Barriers", VLSI Systems and Computations, Computer Science Press, 1981, pp. 143-152.

[4] Mirzaian, A., "Channel Routing in VLSI", Proc. 16th ACM SOTC, 1984, pp. 101-107.

[5] Pinter, R. Y.; "The Impact of Layer Assignment Methods on Layout Algorithms for Integrated Circuits", Ph.D. Thesis, MIT, 1982.

[6] Leiserson, C.E.; Pinter, R. Y.; "Optimal Placement for River Routing", SIAM J. Computing 12 (3), 1983, pp. 447-462.

[7] Blum, M.; Floyd, R. W.; Pratt, V.R.; Rivest, R.L.; Tarjan, R.E.; "Time Bounds for Selection", Journal of computer and Systems Sciences, 7:4, 1972, pp. 448-461.

[8] Aho, A. V.; Hopcroft, J. E.; Ullman, J. D.; "Data Structures and Algorithms", Addison-Wesley, Reading Mass., 1983.

CHANNEL ROUTING WITH SUPERTERMINALS

GUDNI GUDMUNDSSON and SIMEON NTAFOS
Dept. of Computer Science
Univ. of Texas at Dallas
Richardson, Texas

INTRODUCTION

We consider channel routing when the area over a cell is available for routing. This problem was discussed in [5] where there are two layers available inside the channel and one layer (and possibly diffusion underpasses as well) over the cells. Also, Chen's TRIGGER [2] uses three layers inside the channel and up to two layers over the cells. We concentrate on channel routing with two layers available inside the channel and a single layer available over each cell but our approach can be used when more layers are available.

We have proved that the dogleg channel routing problem with two layers inside the channel and a single (infinite capacity) layer over each cell is NP-Hard by reducing it to Szymanski's dogleg channel routing problem [6].

In developing efficient algorithms that produce good solutions, there are two subproblems: a) allocation of connections to the area inside the channel and the areas over the cells on either side, and b) routing of the connections in each area. We expect that heuristics that try to tackle both problems at the same time will not be effective. We propose a three stage approach. First we make the connections over each cell. This produces groups of terminals that are connected. We call these locally connected groups *superterminals*. In the second stage, we select terminals within these superterminals that will be used to completely connect each net inside the channel. Then, in the third stage we use existing channel routers to route the channel.

In this paper we discuss the second stage. We assume that the "outside" connections have been made and we consider the problem of connecting superterminals "inside" the channel. We choose to concentrate on minimizing the density in the channel since most current channel routers route close to density for any kind of channels (i.e. use doglegs or other means to break vertical constraints), e.g. [1] and [7]. We first identify some connections that are necessarily in the optimal solution of the problem. We then give efficient heuristic algorithms to augment this set of connections so that all nets are fully connected.

Connecting the superterminals is similar to the problem considered by Leong and Liu[4]. They give the possibility of some of the terminals in a net being already connected inside the cell as a partial motivation for their problem. However, their solution is only suitable when all the terminals on each side, and in the same net, are connected.

We have done some preliminary evaluations of our three stage approach by applying it to the examples in [5]. In these examples, we used a very simple heuristic based on maximum cardinality matching to determine the connections over the cells. We found that we use fewer tracks over the cells and, in nearly all cases, we also reduce the size of the channel.

ALGORITHMS FOR CONNECTING SUPERTERMINALS

To make our problem more feasible we make the realistic assumption that the superterminals are such that if $i<j<k$ are positions of terminals in a net (on the same side of the channel), and if i and k are in the same superterminal then j is also in that same superterminal, i.e. a terminal can only be preconnected to a neighbor.

The following theorem identifies connections that must be in the optimal solution:

Theorem 1: We can fix connections in the optimal solution as follows:

i) If T and B are two superterminals in the same net, on opposite sides of the channel and are not connected, and there exist terminals t in T and b in B that are in the same column, then $\{t, b\}$ is an optimal connection. The terminals t and b need not be unique.

ii) If ST_1 and ST_2 are two superterminals in the same net and r_1, the rightmost terminal in ST_1, is to the left of l_2, the leftmost terminal in ST_2, and the span of no other superterminal in the net intersects the open interval $]r_1, l_2[$, then $\{r_1, l_2\}$ is an optimal connection.

What remains is to fully connect each net. Connections can be completed by performing a left to right scan and connecting a superterminal to the next superterminal, in the same net, on the "right" (if not already connected).

We propose the following heuristic approach to solving the problem:

Step 1: Make all the connections identified in theorem 1.

Step 2: Use a greedy heuristic to add connections to fully connect all the nets:

a) Scan the channel from left to right.

b) The algorithm is given a target density for the channel. We therefore pack as many connections at any one point as we can without exceeding this density.

c) The decision whether or not to include a connection candidate is made when we reach the end of the candidate. This gives higher priority to short candidates.

There are three levels of sophistication that can be used to make the decision in step 2-c. We can accept a candidate as long as there is room for it, or we can assign priorities to the candidates and only accept a candidate as long as it does not block out a candidate with higher priority. When priority is used it is possible for a candidate to block out another candidate, but be rejected later. We can either ignore this or compensate for it when a candidate is taken out of consideration.

The fact that each terminal gives rise to at most four connection candidates and a careful selection of data structures allows us to implement all the variations of the algorithm in O(area of the channel) time. We use Tarjan's Union-Find method [3] to keep track of the terminals that are connected. To keep track of the density in the channel we only need a linked list of strictly higher "peaks" of local density, going left form the point where we are currently working. Each node stores the rightmost occurrence of that density in the channel so far. Since all active candidates go left from our current point this gives us all the information that we need about the density.

REFERENCES

1. D. Braun et al., "Chameleon: A New Multi-Layer Channel Router". *Proc. 23rd Design Automation Conf.*, pp. 495-502, 1986.

2. H.H. Chen, "TRIGGER: A Three-Layer Gridless Channel Router". *Dig. Tech. Papers, IEEE Int. Conf. on Computer-Aided Design*, pp. 196-199, 1986.

3. E. Horowitz and S. Sahni, *Fundamentals of Data Structures.* Computer Science Press, Potomac, Maryland, 1976.

4. H.W. Leong and C.L. Liu, "A New Cannel Routing Problem". *Proc. 20th Design Automation Conf.*, pp. 584-590, 1983.

5. Y. Shiraishi and J. Sakemi, "A Permeation Router". *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 462-471, May 1987.

6. T.G. Szymanski, "Dogleg Channel Routing is NP-Complete". *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 31-41, Jan. 1985.

7. T. Yoshimura, "An Efficient Channel Router". *Proc. 21st Design Automation Conf.*, pp. 38-44, 1984.

A SUBOPTIMAL SOLUTION TO THE VIA PLACEMENT PROBLEM ‡

TEOFILO F. GONZALEZ and SHASHISHEKHAR KURKI-GOWDARA
Department of Computer Science
University of California
Santa Barbara, CA 93106

ABSTRACT

We consider the via placement problem that arises in multilayer printed circuit board (MPCB) routing. It is shown that this problem can be formulated as a set of integer max-flow problems with max selectors and bundle capacities. Since this integer max-flow problem is an NP-complete problem, we know that there is no known efficient algorithm for its solution. However, a solution to our via placement problem can be obtained by relaxing the integer constraints in the set of integer max-flow problems. The solution to each of the (relaxed) max-flow problems can be obtained by solving a linear programming problem. Our procedure generates a placement with no more than $2 * d_{OPT}$ density, where d_{OPT} is the the density in an optimal placement, in (low) polynomial time. We also show that the column permutation and the board permutation problem can be solved by a similar algorithm to obtain a permutation with density which is not more than $2 * d_{OPT}$. The time complexity of both algorithms is polynomial, however, the time complexity for the second algorithm grows faster than the one for the first algorithm.

INTRODUCTION

Large scale computer systems are built by interconnecting multilayer printed circuit boards (MPCB). Each MPCB consists of a grid with r rows and n columns. The computer chips (components) are mounted on top of the MPCB and their terminals are inserted in drilled-through holes called *pins*. The pins are located at grid points (intersection of a row with a column). Terminals from different chips are connected by printed wires located on any of the layers in the MPCB. Each set of points (pins) that need to be made electrically common is called a *net*. In this paper we use the routing model proposed by So [SO]. In this model only points (pins) on the same line (row or column) can be connected directly. The physical routes for these wires must be confined within the channels on both sides of the line. All the row connections are performed on one layer and all the column connections are carried out on another layer. The main advantage of this model is that the MPCB routing problem is reduced to solving no more than $r + n$ single-line single-layer routing problems. The single-line single-layer routing problem can be solved by the algorithms given in [KKF], [TMK], [RS] and [HS2]. A sample problem is shown in figure 1.1a and a routing is given in figure 1.1b.

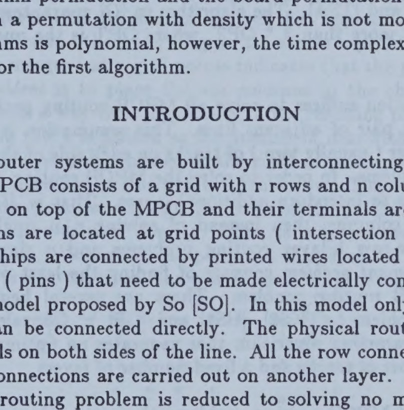
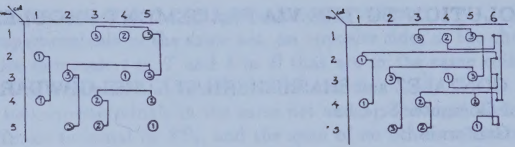

(a) (b)

Figure 1.1: Simple MPCB routing problem.

It is simple to find problem instances that cannot be solved under the above model (figure 1.2a). To solve these problem instances one introduces a set of via columns. These via columns are initially placed to the right of all the pin columns and the vias are located at grid points. Hereafter when we refer to a point we mean a pin or a via; and when we refer to So's model we use the new definition for point. We assume that n also includes the via columns. The vias are assigned to the nets in such a way that a routing under the above model exists (see figure 1.2b).

---

‡ This research was supported in part by the National Science Foundation under Grant DCR - 8503163



(a) MPCB routing problem that cannot be solved without introducing via columns.
(b) A solution to (a) with one via column.
Figure 1.2: MPCB routing problem.

The *via assignment problem* (VAS) consists of adding the least number of via columns and specifying the points (pins and/or vias) that need to be connected directly in such a way that no two pins from the same net are connected to vias from different via columns and a routing under the above model exists. The *multi-via assignment problem* (MVAS) is defined similarly, except that we allow the pins of a net to be connected to vias from more than one via column. As pointed out in [G] finding an efficient approximation solution for the VAS problem is a computationally "difficult" problem, whereas finding a good approximation solution for the MVAS problem is a computationally tractable problem. Approximation algorithms and heuristics for these problems appear in [BK], [G], [TKS1] and [TSA]. The algorithm in [G] guarantees a solution with a number of via columns which is not more than 3 * OPT, where OPT is the number of via columns in an optimal solution.

The above formulation suffices to solve all MPCB routing problems as long as there are enough tracks between each pair of adjacent lines. This assumption is not realistic. In practice, there are only a small number (usually two) of tracks on each side of the line in each of the single-line single-layer routing problems. In order to solve the MPCB routing problems when the number of tracks is small, one needs to introduce additional layers. That is, if there are h layers for the rows and v layers for the columns, then instead of solving r+n single-line single-layer routing problems, we solve r single-row h-layer routing problems and n single-column v-layer routing problems. The *layer assignment problem* consists of finding the least number of layers (h + v) required to solve an MPCB routing problem. There are several heuristics and approximation algorithms to solve this problem ([TKS2], [HS1] and [GS]). Gonzalez and Shashishekhar [GS] introduced a dynamic programming algorithm that generates an optimal solution to this problem when there are a small number of tracks and a fixed number of layers.

One can easily find MPCB routing problems that even when solved optimally under the above model would require an excessive number of layers. A solution with a smaller number of layers can be obtained by moving the via columns and placing them between adjacent chips. This operation is realistic since it can be carried out by just moving the chips enough distance, while maintaining their columnwise relative order. The placement of the via columns is such that the maximum density of any of the single-row routing problems is least possible, where by *density* we mean the maximum number of nets that cross (with a pin to the right and to the left) any point in a row. This problem is called the *via placement optimization problem (VPO problem)*. Note that this placement does not change the column density, however, it may decrease the row densities. Another important observation is that minimizing the row density does not necessarily imply that one can generate a solution with the minimum number of layers. However, minimizing density tends to minimize the number of layers. If we are also allowed to interchange pin columns, the problem is called the *column permutation optimization problem (CPO problem)*. We should point out that the density of an optimal placement for the CPO problem is always smaller than the one for the VPO problem. The CPO problem has been modeled as the *board permutation problem (BP problem)* and heuristics for its solution have been developed by Cohoon and Sahni [CS].

In this paper we study the VPO problem. We present an algorithm that generates a solution with a density which is not more than twice the optimal density d_{OPT}. Our technique consists of reducing the VPO problem to a set of integer max-flow problems with max selectors and bundle capacities. Since this problem has been shown to be NP-complete [S], it is unlikely that one can find an efficient algorithm for its solution. However, if we relax the integer constraints in these max-flow problems, a solution may be obtained by standard linear programming algorithms [GA] or by more recent algorithms [K], [KA]. The solution to the non-integer max-flow problem is not a solution to our VPO problem, however one can obtain a solution to our problem from the solution to the relaxed max-flow problem. In section II we formally define our problem and reduce it to the via placement decision problem. In section III we show that a restricted via placement decision problem can be solved by formulating it as an integer max-flow problem with bundle capacities. In section IV we show that a solution with a density that is within two times the optimal density can be obtained from a solution to the (integer constraint) relaxed max-flow problem. We show in section V how the restrictions made in the previous sections can be handled by a more general algorithm. Therefore, our result holds for the unrestricted VPO problem. In section VI we show how to adapt our algorithm to solve the CPO and the BP problems.

DEFINITIONS AND PROBLEM SIMPLIFICATION

Let P be a multilayered printed circuit board. Board P consists of r rows and n columns (columns 1, 2, ..., n-v contain only pins and the remaining v columns are via columns) for the points (pins and vias). At the intersection of a row with a column there is a point (pin or via). Each point belongs to at most one net. The nets are represented by N_1 , N_2 ,..., N_m and each N_i consists of a set of points that need to be made electrically common. Between every pair of adjacent pin columns there is a channel, and there is a channel to the left (right) of the first (last) pin column. We label the channels with the column number that appears immediately to the left of the channel, if there is one, and zero otherwise. Figure 2.1 shows a problem instance with r = 5, n = 11, v = 4, m = 6, and the integer i at a grid points indicates that the point belongs to net N_i. The objective in the *VPO problem* is to place the via columns in the channels (any number of via columns per channel) in such a way that the density of the resulting problem is least possible. The *point density* of a point p in row k is defined as the number of nets with one point located to the left of point p in row k, and another point located to the right of point p in row k. The *row density* is the maximum of the point densities for the points in that row. The *density of a problem* is defined as the maximum of the row densities over all the rows of the problem. Note that all placements allowed have exactly the same column density. That is why we define the problem density by considering only row densities. We should also point out that each via column has at least two vias assigned to nets. If this is not the case, the via column may be deleted without modifying the connectivity of the pins.

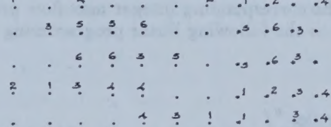


Figure 2.1: MPCB Routing problem.

Instead of solving the VPO problem directly, we solve a set of *via placement decision problems (VPD problems)* and then from their solutions we construct the solution to the optimization problem. The input to the VPD problem is identical to the one for the VPO problem, except that there is another input value denoted by $d_{\max}$. The VPD problem consists of determining whether there is a placement, P, for the via columns with density, denoted by d(P), at most $d_{\max}$. Let d_{curr} be the density of the VPD problem without including the via columns. Clearly, in any placement, P, of the via columns the resulting density d(P) is at most (d_{curr} + v). The decision problem can be solved a logarithmic number of times for $d_{\max}$ between d_{curr} and (d_{curr} + v) until the optimal density bound is found. Thus, by repeatedly solving the decision problem log(v) times, a solution to the optimization problem can be found.

Let us now consider the following three restrictions on a via placement problem.

(R1) If a via in row i is assigned to a net j, then there exists at least one pin from net j in row i.

(R2) If a via in row i is assigned to a net j, then no other via assigned to net j is located in row i.

(R3) If there is a placement P for the via placement problem with density d(P), then there is another placement P' with density d(P') $\leq$ d(P) in which the via columns that appear in each channel are placed in increasing order of their initial column index.

In what follows we use the notation (i,j)-VPO (or (i,j)-VPD) to refer to a VPO (VPD) problem that satisfies restrictions Ri through Rj. It is important to remember that our final result is a polynomial time algorithm for the (unrestricted) VPO problem that generates a solution with density $\leq 2 * d_{OPT}$.

The *integer max-flow problem with bundle capacities* is defined as follows. As input we are given a directed graph G = (N,E) with two special nodes labeled *source* and *sink,* and an edge e_0 directed from the sink to the source; a collection S = $(S_1, S_2, ..., S_l)$ of subsets of arcs (non of which is e_0); and a collection C = $(c_1, c_2, ..., c_l)$ of bundle capacities. Graph G must also satisfy the property that the only edge emanating from the sink and the only edge ending in the source is edge e_0. The problem consists of assigning a non-negative integer flow to each edge in E so as to maximize the flow along e_0 (i.e., from source to sink), provided that flow is conserved at each node (the flow into every node, must equal the flow out of the node) and the bundle capacities are satisfied (for $1 \leq i \leq l$, the sum of the flows assigned to the edges in S_i must not exceed the bundle capacity c_i). The *max-flow problem with bundle capacities* is defined similarly, except that the flows are allowed to be non-negative real numbers.

As an example, consider the integer max-flow problem defined over the directed graph shown in figure 2.2. The edges are labeled with their corresponding flows. The collection S of bundles of arcs is ({ f_1, f_2 }, { f_2, f_3 }, { f_3, f_1 }, { f_4, f_5 }, { f_5, f_6 }, { f_6, f_4 }) and the collection of the bundle capacities is C = (1, 1, 1, 1, 1, 1).

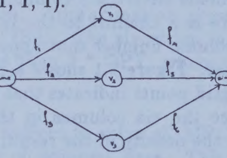


Figure 2.2: Max-flow problem with bundle capacities.

Clearly the maximum integer flow attainable from source to sink, given that the above bundle capacities are to be satisfied, is one. The optimal flows are: either $f_0 = f_1 = f_4 = 1$ and all other f_i's zero; $f_0 = f_2 = f_5 = 1$ and all other f_i's zero; or $f_0 = f_3 = f_6 = 1$ and all other f_i's zero. After relaxing the integer constraints (the flows could be any non-negative real numbers) we obtain a max-flow problem in which the value of the maximum flow is 3/2. The optimal flow is $f_1 = f_2 = f_3 = f_4 = f_5 = f_6 = 0.5$ and $f_0 = 1.5$. Note that the maximum flow from source to sink in an optimal solution for max-flow problem is greater than or equal to the maximum flow from source to sink in an optimal solution for the corresponding integer max-flow problem. The max flow problem defined above can be formulated as the following linear programming problem.

max f_0
subject to
/* conservation rules */
$f_0 - f_1 - f_2 - f_3 = 0$; $f_1 - f_4 = 0$; $f_2 - f_5 = 0$; $f_3 - f_6 = 0$; $f_3 + f_4 + f_5 - f_0 = 0$;
/* bundle capacities */
$f_1 + f_2 \leq 1$; $f_2 + f_3 \leq 1$; $f_3 + f_1 \leq 1$; $f_4 + f_5 \leq 1$; $f_5 + f_6 \leq 1$; $f_6 + f_4 \leq 1$;
/* flow must be a non-negative real */
$f_i \geq 0$

When there are max selectors, each element in a bundle could be the flow on an edge, or the maximum of the flows of several edges. Formally, instead of being given a collection of subset of edges, we are given a collection S of a sets whose elements are subsets of edges (none of which is e_0) and a collection of capacities C. For example S = ({ {a,b,c}, {d,e} }, { {g,h}, {i} }) and C = (x, y). The is interpreted as follows.

max{ f(a), f(b), f(c) } + max{ f(d), f(e) } $\leq$ x; and max{ f(g), f(h) } + f(i) $\leq$ y.

These constraints can be model by introducing variables x_1, x_2 and x_3, together with the following inequalities.

f(a) $\leq x_1$; f(b) $\leq x_1$; f(c) $\leq x_1$; f(d) $\leq x_2$; f(e) $\leq x_2$; $x_1 + x_2 \leq$ x;
f(g) $\leq x_3$; f(h) $\leq x_3$; x_3 + f(i) $\leq$ y;
f(*) ≥ 0; $x_i \geq 0$

Later on it will become clear why we need all of these extensions to the max-flow problem.

FORMULATION OF THE (1,2)-VPD PROBLEM AS AN INTEGER MAX-FLOW PROBLEM

In this section we show that the (1,2)-VPD problem can be formulated as an integer max-flow problem with bundle capacities. However, finding a solution to the above integral flow problem is computationally difficult. In section IV, we show that a solution to the (1,2)-VPD problem can be obtained from an optimal solution to the relaxed max-flow problem, which can be solved in polynomial time. We begin by formulating the (1,3)-VPD problem as an integer max-flow problem, and then we show how to formulate the (1,2)-VPD problem as an integer max-flow problem.

Given a (1,3)-VPD problem, let us show how to solve it by transforming it to the integer max-flow problem. I.e., given a (1,3)-VPD problem X we construct an integer max-flow problem Y such that X has a placement with density $d(P) \leq d_{\max}$ if and only if Y has a flow from source to sink with value = v. First let us construct a *flow graph* from the (1,3)-VPD problem X. There are two special nodes labeled source and sink. For each via column k we introduce the following set of nodes and edges. The set of nodes is $\{ (i, j)_k \mid 1 \leq i \leq 2, 0 \leq j \leq n - v \}$. Assume that these nodes are laid along two parallel horizontal lines, with the node $(1,j)_k$ above $(2,j)_k$ and node $(i,j)_k$ to the left of $(i,j)_k$. The (horizontal) edges in the flow graph are directed from node $(i, j)_k$ to node $(i, j+1)_k$ for all nodes with $1 \leq i \leq 2$, and $0 \leq j < n - v$. The (vertical) edges are directed from node $(1, j)_k$ to node $(2, j)_k$ for $0 \leq j \leq n - v$. Two more edges, one from the node labeled source to the node $(1, 0)_k$, and the other from node $(2, n - v)_k$ to the node labeled sink are added to the flow graph. For example, consider the flow graph in figure 3.1 introduced for the via column labeled 8 with respect to the problem given in figure 2.1. Flow graphs for the other via columns are similar.

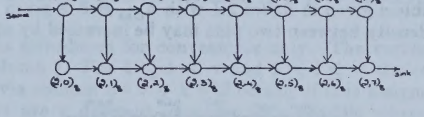
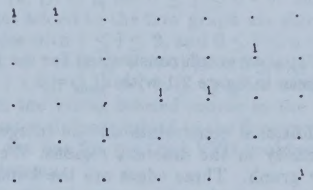
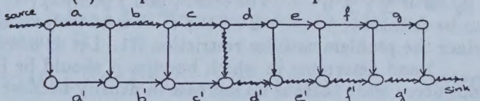


Figure 3.1: Flow graph for via column 8.

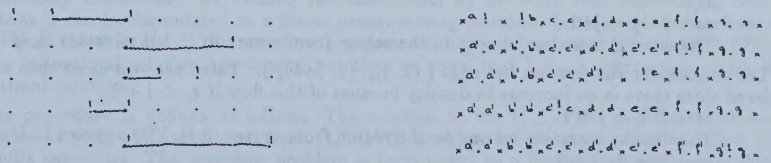
In order for our max-flow problem to model the (1,3)-VPD problem, we need to introduce additional constraints, i.e., more elements in set S. Let us consider the via placement problem with one via column given in figure 3.2a. The flow graph for it is given in figure 3.2b. An optimal flow is illustrated in the same figure as follows: wiggle lines indicate a flow of one, and no wiggle lines indicates a flow of zero. The resulting placement following our convention for the flow in figure 3.2b is given in figure 3.2c.



(a) Via placement decision problem.



(b) Flow graph for the via column.



(c) Placement for (a) obtained from the flow in (b). (d) density increase defined by the flow.

Figure 3.2.

In figure 3.2c one may observe that because of the placement of the via column, the density in the regions depicted by wiggle lines is increased by one. Clearly, the placement of a given via column will increase the density in certain places. The flow along the edges in figure 3.2b is specified by the letters a-g and a'-g'. In figure 3.2d we show the regions in the placement where the density increases, because of the flows a-g and a'-g'. Clearly, there is a limit on the amount of density that a region can absorb. The amount by which the density of a region can increase is given by the following graph.

Given a via placement decision problem P, we construct G_v (*capacity graph*) as follows:

The set of vertices is define by the set of grid points { (i, (j, k)) | $1 \leq i \leq r$, $0 \leq j \leq$ n-v, and $0 \leq k \leq v$ }.

The edges in the graph are defined as follows. For $1 \leq i \leq r$, $0 \leq j \leq$ n-v, and $0 \leq k < v$, there is a directed edge from vertex (i, (j, k)) to vertex (i, (j, k+1)). For $1 \leq i \leq r$ and $0 \leq j <$ n - v, there is a directed edge from vertex (i, (j, v)) to vertex (i, (j+1, 0)).

The vertex (i, (0,0)) is introduced for convenience only. The vertex (i, (j, 0)) corresponds to the pin located in row i column j. For $1 \leq k \leq v$, the column (*, (j, k)) indicates the location that via column n - v + k will occupy if it is placed in channel j. Let d(i,j) = the current density of row i from column j to column j+1, for $1 \leq j <$ n-v, and let d(i,0) = d(i,n-v) = 0. The capacity for the edge with tail at vertex (i, (j, k)) is $d_{\max}$ - d(i,j). The capacity of an edge indicates that after the placement is performed, the density of the edge might increase by that amount. The capacity graph for the via placement problem given in figure 2.1 with $d_{\max} = 3$ is given by figure 3.3. The capacity graph indicates that the density between two pins may be increased by at most that amount.

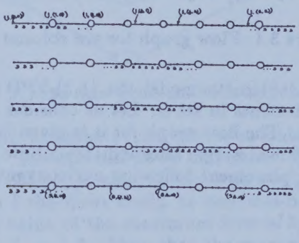


Figure 3.3: Capacity graph constructed for the problem given in figure 2.1 with $d_{\max} = 3$.

Let us now formalize the additional constraints on the integer max-flow problem that will limit the additional increase in density in the different regions. We will add flows from the flow graph to the edges in the capacity graph. These edges are the bundles and their sum must be at most the capacity of the edge. Let us consider via column k with its flow graph defined over the vertices $(1, j)_k$ and $(2, j)_k$ for $0 \leq j \leq$ n - v. The start (end) point s_i (e_i) is the leftmost (rightmost) pin that needs to be connected to the via on the i^{th} row of via column n - v + k. Remember that such points exist since the problem satisfies restriction R1. Let us consider the flow on the edge ($(1, j)_{n-v+k}$, $(1, j+1)_{n-v+k}$) and determine in which bundles it should be included. There are two cases that need to be considered since there is no increase in density because of this flow if $e_i >$ j+1.

case 1: $e_i =$ j+1.

The density increases by one on the region from vertex (i, (j+1, 0)) to vertex (i, (j+1, k)).

case 2: $e_i <$ j+1.

The density increases by one on the region from vertex (i, (j, k)) to vertex (i, (j+1, k)).

Let us consider the flow on the edge ($(2, j)_k$, $(2, j+1)_k$). There are two cases that need to be considered since there is no increase in density because of this flow if $s_i <$ j + 1.

case 1: $s_i =$ j + 1.

The density increases by one on the region from vertex (i, (j, k)) to vertex (i, (j+1, 0)).

case 2: $s_i >$ j + 1.

The density increases by one on the region from vertex (i, (j, k)) to vertex (i, (j+1, k)).

The bundles are obtained from the set of flows that affect each edge and the capacity is the capacity of the corresponding edge in the capacity graph. At this point it is simple to prove the following lemma.

Lemma 3.1: The (1,3)-VPD problem X has a placement P with density d(P) $\leq d_{\max}$ iff the integer max-flow problem Y constructed from it has a flow from source to sink equal to v.

Proof: By the above discussion. □

If there is a solution to this integral flow problem with the bundles with a flow from source to sink equal to v, we know that the (1,3)-VPD problem has a solution. As we mentioned before, the max-flow problem that we constructed is an instance of a problem which has been shown to be NP-hard. However, if we relax the integer constraints on the flows it can be formulated as a linear programming problem and solved in polynomial time.

Now let us now relax the "ordering" assumptions that we had made on placing more than one via column in the same channel. That is, there might not be an optimal placement with the property that all the via columns assigned to the same channel are placed in order of their initial column index. Remember that we refer to this problem as the (1,2)-VPO or the (1,2)-VPD problem. To deal with this situation we modify the capacity graph by introducing additional nodes and edges.

The capacity graph G_v is now defined as follows:

The set of vertices is defined by the set of grid points { (i, (j, k)) | $1 \leq i \leq r$, $0 \leq j \leq n-v$, and $0 \leq k \leq v^2$ }.

The edges in the graph are defined as follows. For $1 \leq i \leq r$, $0 \leq j \leq n-v$, and $0 \leq k < v^2$, there is an edge from vertex (i, (j, k)) to vertex (i, (j, k+1)). For $1 \leq i \leq r$ and $0 \leq j < n-v$, there is a directed edge from vertex (i, (j, v^2)) to vertex (i, (j+1, 0)).

The vertex (i, (0,0)) is introduced for convenience only. The vertex (i, (j, 0)) corresponds to the pin located in row i column j. For $1 \leq k \leq v$ and $1 \leq l \leq v$, the column (*, (j, (l-1)v + k)) indicates the location that via column n - v + k will occupy if it is assigned to its l^{th} place in channel j. Note that now there are v different locations in a channel where each via column can be placed. This allows arbitrary placement orders of the via columns in each channel. Let d(i,j) = the current density of row i from column j to column j+1, for $1 \leq j < n-v$, and let d(i,0) = d(i,n-v) = 0. The capacity for the edge with tail at vertex (i, (j, k)) is $d_{\max}$ - d(i,j). The capacity of an edge indicates that after the placement is performed, the increased density in the edge must be less than or equal to it.

The construction of the flow graphs is also modified as follows. There are two special nodes labeled source and sink. For each via column k we introduce the following set of nodes and edges. The vertices $(1, (j, t))_k$ and $(2, (j, t))_k$ for $0 \leq j \leq n - v$, and $1 \leq t \leq v$ are added to the flow graph. The (horizontal) edges added to the flow graph are directed from vertex $(i, (j, t))_k$ to vertex $(i, (j, t+1))_k$ for all vertices with $1 \leq i \leq 2$, and $0 \leq j \leq n - v$ and $1 \leq t < v$. We also add the (horizontal) edges from (i, (j, v)) to (i, (j + 1, 0)) for $1 \leq i \leq 2$ and $0 \leq j < n - v$. We also add (vertical) edges directed from $(1, (j, t))_k$ to $(2, (j, t))_k$ for vertices with $0 \leq j \leq n - v$ and $1 \leq t \leq v$. Two more edges, one from the vertex labeled source to the vertex $(1, (0, 1))_k$, and the other from $(2, (n - v, v))_k$ to the vertex sink are added to the flow graph. The bundles are defined by following a procedure similar to the one for the (1,3)-VPD problem. From our construction rules it is simple to prove the following lemma.

Lemma 3.2: The (1,2)-VPD problem X has a placement P with density $d(P) \leq d_{\max}$ iff the max-flow problem Y constructed from it has a flow from source to sink equal to v.
Proof: By the above discussion. □

SUBOPTIMAL SOLUTION FOR THE (1,2)-VPO PROBLEM

In the previous section, we presented a method for solving the (1,2)-VPD problem by reducing it to the integer max-flow problem with bundle capacities. Since the integer max flow problem is computationally intractable, we relaxed the constraints by allowing real numbers as flows, and showed that it can be formulated as a linear programming problem. In this section we show how to generate from the solution to the max-flow problem a suboptimal solution for the (1,2)-VPO problem. The suboptimal solution has density less than or equal to 2 * d_{OPT}, where d_{OPT} is the density of an optimal solution.

Our procedure is defined as follows. The solution to the (1,2)-VPO problem is obtained by solving a set of (1,2)-VPD problems. Each (1,2)-VPD problem is formulated as a max-flow problem with bundle capacities. The max-flow problem is formulated as a linear programming problem and solved. At this point we obtain a pseudo-placement with density $d_{\max}$ and we know that there is no placement for the via placement optimization problem with density less than $d_{\max}$. Hence, $d_{\max} \leq d_{OPT}$. Now the solution to the LP problem provides us with the flows for the flow graphs. If it is the case that all the flows are integers, then we are done, and we have a placement with density $d_{\max} = d_{OPT}$. In general this is not going to be the case. For example consider the flow graph for via column k, given in figure 4.1. The value assigned to an edge is the flow for that edge in our solution.

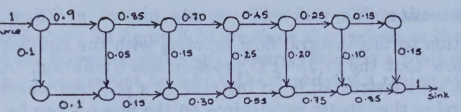


Figure 4.1: Non-integral solution for the flow graph of via column k.

If we follow our old interpretation it seems to suggest that fractions of the via column are to be placed in different channels. This is not allowed in our via placement problem. However, from the structure of the flow graph we know that there is at least one path from the source to the sink that goes through the flow graph for via column k with the property that each horizontal edge has a flow greater than or equal to 0.5. Any one of these paths is selected to generate our solution to the via placement problem, i.e., the via column will be placed at the position indicated by the vertical edge in this path.

Now the question is what is the density of the placement obtained this way? The answer is simple. With the non-integral flow, the density of the pseudo-placement was at most $d_{\max}$. Since we have at most doubled the flow of each edge, then the density of the placement that we obtain is at most $2 * d_{\max} \leq 2 * d_{OPT}$.

Lemma 4.1: Our algorithm generates a placement for the (1,2)-VPO problem with density $\leq 2 * d_{OPT}$.
Proof: By the above discussion. □

APPROXIMATION ALGORITHM FOR THE VPO PROBLEM

In the previous sections we presented an algorithm that generates an approximate solution to the (1,2)-VPO problem with a density which is at most two times the density in an optimal solution. However, we assumed that the via placement problem satisfied restrictions R1 and R2. In this section we show how to generate a suboptimal solution for the via placement problem without these restrictions. These restrictions will be overcome by the introduction of max selectors. These max selectors will not affect our approximation bound, they will just increase the time complexity bound.

We now show how to solve the (1,1)-VPD problem by introducing max selectors. Suppose that three via columns on some row i have a via assigned to the same net. Assume that by using our previous algorithm we generate the placement given by figure 5.1.

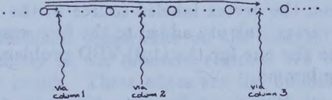


Figure 5.1: Placement

The wiggle lines indicate where the densities increase by one. The problem is that the density our procedure computes will be increased by three in some locations. This is incorrect, it should be at most one. This problem can be handled by selecting the maximum of the three flows. Hence the use of max selectors. It is simple to verify that restriction R2 can be eliminated by the introduction of appropriate max selectors.

Lemma 5.1: The (1,1)-VPD problem X has a placement P with density $d(P) \leq d_{\max}$ iff the max-flow problem Y constructed from it has a flow from source to sink equal to v.
Proof: By the above discussion. □

Now let us consider restriction R1, i.e., there might be several vias located on different via columns that belong to the same net, but such a net does not have a pin in that row. Note that whenever this occurs there must be at least two vias assigned to this net. For the moment let us assume that there are three such vias. Figure 5.2 shows the flow graphs for the three via columns containing the three vias. The wiggle lines show the edges with flow equal to one.

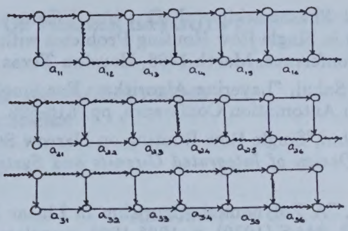


Figure 5.2: Flow graph for the three vias

In the previous algorithm there is no provision to handle this problem because we add flows to bundles depending on s_i and e_i. Now there are rows with no s_i nor e_i. If we try to solve the problem using the previous algorithm without making any additions to the bundles, we will obtain infeasible solutions. We now show how to handle these situations by introducing max selectors. It is simple to show that the density between the i^{th} pair of adjacent horizontal nodes is increased by at most

$$\max \{ f_{i,1} - f_{i,2}, f_{i,2} - f_{i,3}, f_{i,3} - f_{i,1}, f_{i,1} - f_{i,3}, f_{i,3} - f_{i,2}, f_{i,2} - f_{i,1} \}.$$

By using this expression appropriately, we can easily compute the increase in the density of each edge in the capacity graph. Note that the max selectors are not simple flows, they are linear expressions involving flow variables. The formulation as a linear programming problem in this case is similar to the method discussed in section II. It is simple to prove the following theorem, which is our main result.

Theorem 5.1: The VPD problem X has a placement P with density $d(P) \leq d_{\max}$ iff the max-flow problem Y constructed from it has a flow from source to sink equal to v.
Proof: By the above discussion. □

Corollary 5.1: Our algorithm generates a placement for the VPO problem with density $\leq 2 * d_{OPT}$.

DISCUSSION

It is simple to verify that the linear programming problems required to solve the VPO problem have a number of variables and equations which are bounded above by a polynomial in n. Also, the amount of space required to represent each linear programming problem is bounded above by a polynomial in n. Since the more recent algorithms which solve linear programming problems take time which is polynomial in the problem size, we know that the total time complexity for our algorithm is polynomial. For brevity, we do not include a sharper time complexity bound.

Our algorithm can be easily adapted to solve the CPO problem and the BP problem. The time complexity of the algorithm grows faster than the one for the previous algorithm, however, it is bounded by a polynomial in n. Also, for brevity, the details are omitted.

In many applications of the CPO problem, we might require two adjacent columns to be remain adjacent in any permutation. This additional restriction can also be handled by our algorithm. In practice, the VPO problem is defined in such a way that one does not allow via columns to be placed in certain channels. This additional restriction can also be easily handled by our algorithm.

REFERENCES

[BK] Bhasker, J. and S. Sahni, "Via Assignment in Single Row Routing," Proceedings of the Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, Springer-Verlag, Dec. 1986, 154-176.

[CS] Cohoon, J. and S. Sahni, "Heuristics for the Board Permutation Problem, Proceedings of the 1983 IEEE ICCAD Conference, 81-83.

[GA] Gass, S.,"Linear Programming," Mc-Graw Hill, 1969.

[G] Gonzalez T., "An Approximation Algorithm for the Via Assignment Problem," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. CAD-3, No 4, Oct. 1984, pp 257-264.

[GS] Gonzalez, T. and Shashishekhar, K.G., "An Efficient Algorithm to Minimize the Number of Layers in Single Row Routing Problems with Fixed Street Capacity," 1986 Physical Design Conference, March 1986, Houston Texas.

[HS1] Han S. Y. and S. Sahni, "Layering Algorithms For Single Row Routing," Proceedings of the 22nd Design Automation Conference, pp. 516-522.

[HS2] Han, S. and S. Sahni,"Single Row Routing on Narrow Streets," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* CAD-3, No.3, July 1984, pp 235-241.

[K] Khachiyan, L. G., "A Polynomial Algorithm in Linear Programming," *Doklady Akademiia Nauk SSSR,* 244:S (1979), p. 1093-1096, translated in *Soviet Mathematics Doklady,* 20:1 (1979), pp. 1-12.

[KA] Karmarkar, N., "A New Polynomial-Time Algorithm for Linear Programming," Proceedings of the 16th Symposium on Theory of Computing, April 1984, 302-311.

[KKF] Kuh E., T. Kashiwabara and T. Fujisawa, "On Optimal Single-Row Routing", *IEEE Transactions on Circuits and Systems,* Vol. CAS-26, No. 6, June 1979.

[RS] Raghavan R. and S. Sahni, "Optimal Single Row Router", *IEEE Transactions on Computers,* Vol. C-32, #3, March 1983, pp. 209-220.

[S] Sahni, S., "Computationally Related Problems," *SIAM Journal on Computing,* 3. 4 (Dec. 1974), pp. 262-279.

[SBR] Sahni, S., Bhatt, A. and Raghavan, R., "The Complexity of Design Automation Problems," Proceedings of the 19th Design Automation Conference, June 1981.

[SO] So H., "Some Theoretical Results on the Routing of Multilayer Printed-Wiring Boards," IEEE Symposium on Circuits and Systems 1974, pp. 296-303.

[TKS1] Ting B., E. Kuh and A. Sangiovanni-Vincentelli, "Via Assignment Problem in Multilayer Printed Circuit Board", *IEEE Transactions on Circuits and Systems,* Vol. CAS-26, No. 4, April 1979.

[TSA] Tsukiyama S., I. Shirakawa and S. Asahara, "An Algorithm for the Via Assignment Problem in Multilayer Backboard Wiring", *IEEE Transactions on Circuits and Systems,* Vol. CAS-26, No. 6, June 1979.

[TKS2] Tsukiyama, S., E. S. Kuh and I. Shirakawa, " On the Layering Problem on Multilayer PWB Wiring," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. CAD-2, No.1, January 1983, pp. 30-38.

THREE-LAYER WIRABILITY OF PLANAR LAYOUTS ‡

TEOFILO F. GONZALEZ and SI-QING ZHENG
Department of Computer Science
University of California
Santa Barbara, CA 93106

ABSTRACT

It is well known that every planar layout is four-layer wirable. If we decrease the number of layers, there are planar layouts that cannot be wired on three layers. The problem of determining whether a planar layout is three-layer wirable is an NP-complete problem. A planar layout may be stretched vertically (horizontally) by introducing empty rows (columns). Clearly, stretching a planar layout increases its area; however, if it is stretched in certain locations it can be wired in fewer than four layers. In this paper we consider two layout stretching schemes. We present lower bounds for the worst case wiring area bounds under these two stretching schemes. We show that our lower bound for the two-layer wiring is tight and our lower bounds for three-layer wiring are not too far from known upper bounds for this problem.

I. INTRODUCTION

The *rectangle routing problem (RRP)* is a fundamental problem in VLSI design automation. As input we are given a rectangular *grid R* determined by the horizontal lines with y-coordinate values i, $0 \le i \le h + 1$ (called *tracks* or *rows*) and the vertical lines with x-coordinate values j, $0 \le j \le w + 1$ (called *columns*). The horizontal lines with y-coordinate values 0 and h + 1 and the vertical lines with x-coordinate values 0 and w + 1 form the *boundary* of R. Let $N = \{N_1, N_2, ..., N_p\}$, where each N_i is a subset of grid points on the boundary of R (excluding the corners of R), such that $N_i \cap N_j = \emptyset$ for all $i \neq j$. Each set N_i is called a ***net*** and its grid points are called *terminals.* We assume that there are k *conducting layers* $L_1, L_2, ..., L_k$, each is a copy of the channel grid, and L_{i+1} is considered to be laid upon L_i, $1 \le i \le k - 1$. Contacts between two layers (*vias*) can be introduced only at grid points. Under the *knock-knee* model a k-layer wiring (which is the final routing solution) is a three dimensional structure which can be characterized by two mappings: wire layout and layer assignment. A *wire layout* for a RRP is a mapping that associates each net N_i to a connected subgraph W_i of the grid R such that W_i does not share an edge with W_j for all $j \neq i$. This wire layout is called a *planar layout* or a *path disjoint layout* or simply a *layout.* We use $W = \{W_1, W_2, ..., W_p\}$ to denote the wire layout. The *layer assignment* of a planar layout is a mapping that associates each edge in W to a layer in $\{L_1, L_2, ..., L_k\}$ in such a way that for any W_i and W_j, $i \neq j$, if edges (p_1, p_2), (p_2, p_3) in W_i are assigned to L_s and L_t, respectively, and $(p_2, p_4) \in W_j$ is assigned to L_u, then $u > \max\{s, t\}$ or $u < \min\{s, t\}$. A solution for an RRP is a *k-layer wiring* formed by the composite mapping of wire layout and layer assignment. Obviously, in a wiring the segments of the same wire W_i can be connected through a via without sharing a grid point with a segment of another wire W_j in any layer. Physically speaking, in a wiring all terminals from the same net are made electrically common and no two distinct nets are connected. In practice knock-knee wirings minimize crosstalk since the area of the grid shared by two different nets is limited to grid points. A variation of the rectangle routing problem is the *channel routing problem (CRP)* in which all terminals appear on the top and bottom sides of the grid R and the objective is finding a k-layer wiring with the least number of tracks.

The above characterization of the wiring provides an approach for constructing a wiring for a CRP by finding a planar layout with least number of tracks for the given CRP and then finding a layer assignment for the layout. This approach is used by the routing algorithm for the two-terminal net CRP (each net is of size 2) by Preparata and Lipski ([PL]). The first phase of their algorithm finds a minimum-track planar layout that satisfies some special properties. In the second phase of their algorithm, the planar layout is transformed into a three-layer wiring by a powerful transformation (legal partition of the diagonal diagram induced by the planar

---

‡ This research was supported in part by the National Science Foundation under Grant DCR - 8503163. A preliminary version of this paper appears as UCSB technical report TRCS87-11 and TRCS86-26.

layout$_c$). Their algorithm guarantees a three-layer optimal solution. Several other routing algorithms for the CRP are also based on this approach (e.g., see [PS], and [SP]).

Brady and Brown ([BB]) showed that any planar layout can be transformed into a four-layer wiring with dimensions identical to the dimensions of the planar layout. The implication of this result is that one can reduce the rectangle routing problem to the problem of finding a planar layout, since for any planar layout a four-layer wiring is always possible in the layer assignment phase. One may consider this two-phase four-layer routing approach as "standard". For example, necessary and sufficient conditions for the existence of a planar layout for the two-terminal net RRP are given in [F]. If these conditions are met, a planar layout can be found by the algorithms in [F] and [MP]. A four-layer wiring for this planar layout can be found by applying the layer assignment algorithm given in [BB]. It is not known whether the layouts generated by the algorithms in [F] and [MP] are three-layer wirable. By using the reduction given in lemma 2.1 (refer to the next section) Lipski ([Li]) gives a 19-row wire layout that is not three-layer wirable. He also shows that the problem of deciding whether a given planar layout is three-layer wirable is an NP-complete problem.

A planar layout may be *stretched vertically (horizontally)* by introducing between a pair of adjacent rows (columns) an empty row (column) (figure 3.1) without a horizontal (vertical) wire. Clearly, stretching a planar layout increases its area; however, if it is stretched in appropriate places it can be wired in fewer than four layers. Let us now investigate the trade-off between the routing area and the number of layers needed for wiring a planar layout. Let A(W) denote the area of planar layout W. The simple stretching algorithm described in [MP] generates a two-layer wiring with area not larger than 2 * A(W), by vertically stretching it between every pair of adjacent rows. In this paper we propose two different stretching schemes. Then, we show that for each of these schemes if two conducting layers are available the area bound of 2 for arbitrary planar layouts is tight, i.e. there exist planar layouts with area A(W) which cannot be stretched and wired in an area less than $(2 - \varepsilon) * A(W)$, for all $\varepsilon > 0$. Before we consider the three-layer wirability of planar layouts, we classify planar layouts into four classes according to their structure. Then, for each layout class and each stretching scheme we give lower bounds for the worst case wiring area.

II. PRELIMINARIES

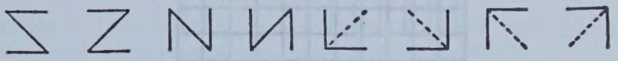
In this section we review some definitions and results from [PL]. Their algorithm for finding a three-layer wiring for a given planar layout W consist of the following steps:

(i) construct the diagonal diagram D that corresponds to the layout W;
(ii) find a legal partition P of D that partitions D into two-colorable regions; and
(iii) find a layer assignment W' of W from P.

The *diagonal diagram* corresponding to the given layout is constructed as follows. At each grid point in R where W has a bend, a $(\sqrt{2}/2)$-length diagonal (called *half diagonal*) emanating from the grid point and internally bisecting the bend wire is introduced. Thus, in case there is a knock-knee at the grid point, a $\sqrt{2}$-length diagonal (called *full diagonal*), centered at the grid point, is formed. We assume that two wire segments from the same wire can form a knock-knee. One may easily transform layouts with these knock-knees and loops. We elaborate on layout transformations in the discussion section. The resulting geometric structure from this transformation is called a *diagonal diagram.* The *core* diagonal diagram of a given layout is the diagonal diagram with the half-diagonals deleted (figure 2.2). For the grid R, the partition grid G(R) is defined as follows. The *grid points* of G(R) are the points $(x + 1/2, y + 1/2)$, $0 \leq x \leq n$, $0 \leq y \leq m$. The grid points with $x = 0$, $x = n$, $y = 0$ or $y = m$ are called *boundary* points and the other points are called *internal* points. A vertical (horizontal) *grid line* in G(R) is the smallest line segment that includes all the grid points with the same y-coordinate (x-coordinate). The space between any two adjacent horizontal (vertical) grid lines is called a *row (column)* of G(R). Note that a row (column) in R is a horizontal (vertical) grid line in R, but a row (column) in G(R) is all the space between two adjacent horizontal (vertical) grid lines in G(R). The *edges* of G(R) are the segments connecting each point with its immediate neighbors, vertically, horizontally, or at 45-degree angles. It should be noted that a full diagonal in D is an edge of G(R), the end points of a full diagonal are grid points of G(R) and no two full diagonals in D cross. Let D denote the core diagonal diagram of layout W. The end points of the diagonals of D lying on nonboundary grid points of G(R) are called *vertices* of D. We say that

D has degree i, $1 \le i \le 4$, at vertex (s, t) if there are i full diagonals with end points at (s, t). We say that D is of *degree* i, $1 \le i \le 4$, if the maximum degree of any vertex in D is i; otherwise the degree of D is zero. A *legal partition* P of D (figure 2.2) is any collection of edges in G(R) satisfying the following conditions:

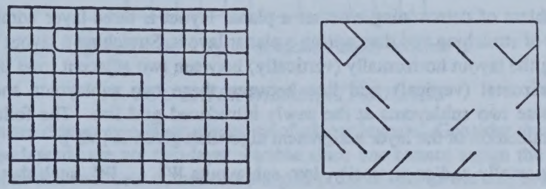
(a) Every internal point of G(R) is incident with an even number of edges in P;
(b) The diagonals in P are exactly the diagonals in D; and
(c) P does not contain any of the patterns shown in figure 2.1.
(dashed lines mean that the diagonal must not be present).



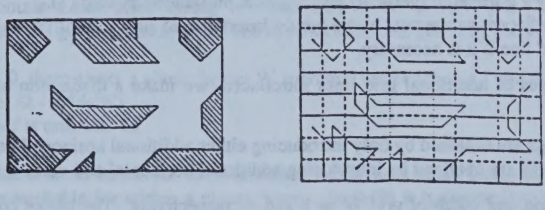
Forbidden patterns.
figure 2.1

The following lemma shows the significance of the notion of diagonal diagram D constructed from W and the legal partition P with respect to D.

Lemma 2.1: ([PL]): If there is a legal partition P of the core diagonal diagram D induced by the planar layout W, then there exists a three-layer assignment for W. Furthermore, a three-layer assignment for W can be easily constructed from P.



Planar Layout   Core Diagonal Diagram



Legal Partition   3-Layer wiring

top layer (solid lines); middle layer (dotted lines) bottom layer (dashed lines)
figure 2.2: Examples.

We omit the description of the process for finding a layer assignment of W from a legal partition P of D. Interested readers should refer to [PL] for details. In figure 2.2 we give a layout W, its corresponding core diagonal diagram D, the legal partition P of D and the three-layer wiring for W obtained from P. In the remaining portion of this paper we use the term diagonal diagram to refer to a core diagonal diagram.

The previous lemma allows us to reduce the problem of three-layer wirability of a planar layout W to the problem of determining the existence of a legal partition P with respect to W. The crucial notion in this reduction is that of full wire layout. We say that a layout W is a *full layout* if every nonboundary edge in R is covered by a wire in W. For example, the full layout constructed from the layout in figure 2.2 is given in figure

2.3. We say that a wire layout W contains a *loop* if there is a path ($(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)$) in W such that $k > 4$ and $v_1 = v_k$, where $v_1, v_2, \ldots, v_k$ are grid points in R. If we only consider two-terminal net full layouts (we call the full layout of this type *standard full layout*) and allow wires with loops, then there is one-to-one correspondence between standard full layouts in R and diagonal diagrams in G(R).

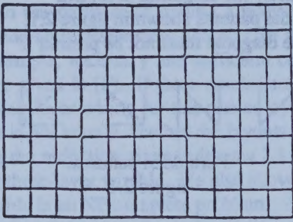


figure 2.3: Full layout for the planar layout given in figure 2.2.

Theorem 2.1: ([L]) A loop-free standard full wire layout W^* in R is three-layer wirable iff there exists a legal partition P of D in G(R) with respect to W^*.

III. LAYOUT STRETCHING SCHEMES

Since the problem of determining whether a planar layout is three-layer wirable is NP-complete, we investigate the problem of stretching and then wiring a planar layout. Stretching a layout vertically (horizontally) is equivalent to dividing the layout horizontally (vertically) between two adjacent rows into two sublayouts, then inserting an empty horizontal (vertical) grid line between these two sublayouts and merging the vertical (horizontal) wires in these two sublayouts at the newly introduced grid line. The following layer assignment scheme is a direct generalization of the layer assignment algorithm given in [MP].

(1) Divide W horizontally and/or vertically, into sublayouts $W^1, \ldots, W^t$, such that W^i, $1 \le i \le t$, is three-layer wirable;
(2) Find a k-layer wiring for each sublayout W^i of W;
(3) Extend the grid R to form grid R' by inserting a horizontal (vertical) empty grid line between sublayouts separated by a horizontal (vertical) dividing line introduced by step (1).
(4) Merge two adjacent sublayouts at the newly inserted grid line and introduce vias at the grid points on the new grid line if it is necessary.

Depending on the types of additional grid lines introduced, we make a distinction between the following two stretching schemes:

Scheme I: Wirings are obtained by only introducing either additional horizontal or vertical grid lines.
Scheme II: Wirings are obtained by introducing additional horizontal and vertical grid lines.

Let the height and width of grid W be h and w, respectively. The layout corresponding to the wiring obtained by the above algorithm is denoted by W'. Clearly, for Scheme I we have

$$A(W') = (1+t/h) * A(W), \quad (\text{or } A(W') = (1+t/w) * A(W))$$

where t is the number of additional horizontal (or vertical) grid lines. For Scheme II we have

$$A(W') = (1+t/h)*(1+u/w) * A(W),$$

where t and u are the number of additional horizontal and vertical grid lines introduced, respectively.

Stretched layouts under scheme I and II for the layout in figure 2.2(a) given in figures 3.1 (a) and 3.1 (b), respectively.

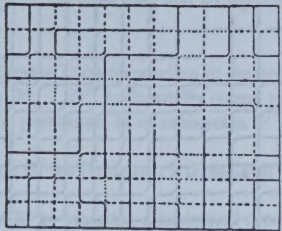


figure 3.1 (a): Layout stretched under scheme I.

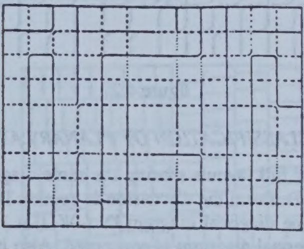


figure 3.1 (b): Layout stretched under scheme II.

IV. TWO-LAYER WIRABILITY BOUNDS

In this section we discuss two-layer wirability of planar layouts. Consider the simple layouts given in figure 4.1. Both of these layouts are not two-layer wirable since one cannot assign the wire segments in these layouts so that wires from different nets do not share a grid point in one of the two available layers. Consider the full planar layout W, which has identical dimension in both directions, shown in figure 4.2. It is simple to verify that any sublayout consisting of two adjacent rows (columns) contains a sublayout with the structure shown in figure 4.1. Hence, under scheme I no less than h-1 (w-1) additional horizontal (vertical) grid lines are needed in order to obtain a two-layer wiring W'. The area of W' is $A(W') \geq (2 - 1/h)*A(W) = (2 - 1/w)*A(W)$. A straight forward generalization of the above observations is given by theorem 4.1.

Theorem 4.1: For any $\varepsilon > 0$, there exists a planar layout W such that under scheme I and II any two-layer wiring W' for it has area $A(W') > (2 - \varepsilon)*A(W)$.
Proof: For brevity the proof is omitted. □

The above theorem provide a lower bound for the worst case wiring area under scheme I and scheme II when two conducting layers are available for wiring a planar layout. In [MP] it is shown that by inserting an empty grid line between any two adjacent rows of a given planar layout, a two-layer wiring can be constructed. Therefore, the area bound 2 for wiring planar layouts under scheme I and II is best possible.

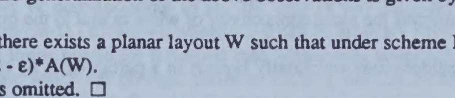


figure 4.1

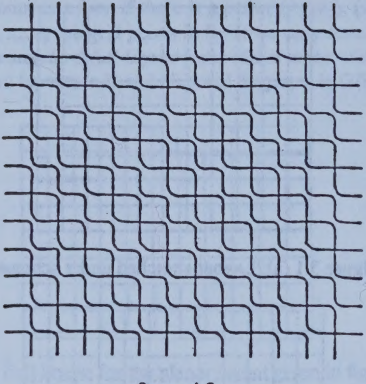


figure 4.2

V. A CLASSIFICATION OF PLANAR LAYOUTS

Let $W^*(D)$ denote the set of full layouts sharing the same diagonal diagram D and $S^*(D)$ denote the standard full layout corresponding to D. From the discussion in section II we know that for a given wiring grid R, all layouts in R that share the same diagonal diagram D ($W^*(D)$) are equivalent when obtaining three-wirings through legal partitions of diagonal diagrams is concerned. Also it is easy to verify that for any diagonal diagram D if $S^*(D)$ contains a loop, so does any other full layout in $W^*(D)$. Because of this, we use the standard full layout $S^*(D)$ as the representative of $W^*(D)$.

When we explore the three-layer wirability of planar layouts, we need to consider the following two aspects of planar layouts. First, we believe that the three-layer wirability of planar layout depends on the structural complexity of layouts. For example, the layout shown in figure 4.1 is not two-layer wirable. However, it is straight forward to construct a three-layer wiring for this layout. When three-layer wiring is concerned, the structure of this layout is simple. Second, we believe that in practice there are layouts with certain structures that may never appear. For example, in a layout corresponding to a diagonal diagram of degree three or degree four there are loops. Those loops corresponding to the vertices of degree three and four in the diagonal diagram can be easily eliminated by slightly modifying the layout so that the modified diagonal diagram has degree less than 2 and the modified layout maintains the same connectivity of wires as that in the original layout.

Based on these considerations we classify layouts in a particular R by their core diagonal diagrams D into four classes:

Class (a): layouts whose core diagonal diagrams are of degree greater than two;

Class (b): layouts whose core diagonal diagrams are of degree two and the standard full layouts corresponding to their core diagonal diagrams contain loops;

Class (c): layouts whose core diagonal diagrams are of degree 2 and the standard full layouts corresponding to their core diagonal diagrams are loop-free; and

Class (d): layouts whose core diagonal diagrams are of degree one.

In the following sections we investigate the three-layer wirability under scheme I and scheme II for each of these layout classes.

VI. THREE-LAYER WIRABILITY BOUNDS FOR SCHEME I

The technique used to deriving lower bounds for the worst case wiring area under scheme I and scheme II in section IV can be generalized to handle the case when there are three conducting layers available for wiring. If we are restricted to constructing a three-layer wiring by finding a legal partition of the diagonal diagram corresponding to the given layout, the stretching and wiring schemes given in section III can be restated as follows.

(1) Construct the core diagonal diagram D corresponding to the given layout W on the grid R;

(2) Divide D on G(R) into blocks $D^1, ..., D^t$, horizontally or/and vertically, by partitioning lines on grid G (accordingly layout W on R is partitioned in to sublayouts $W^1, ..., W^t$, such that W^i, $1 \le i \le t$, is three-layer wirable);

(3) Find legal partition P^i for each D^i and construct a three-layer wiring A^i for the sublayout W^i from P^i of D^i;

(4) Extend the grid R to form grid R' by inserting a horizontal (vertical) empty grid line between sublayouts separated by a horizontal (vertical) division line introduced in step (2).

(5) Merge two adjacent sublayouts in the division at the newly inserted grid line and introduce vias at the grid points on the new grid line if it is necessary.

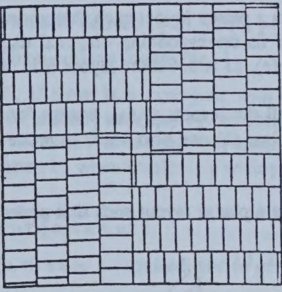


figure 6.1: Strip tile arrangement A_{11}

Suppose there exists an m-row by n-column diagonal diagram D, where $1 < m < n$, for which there is no legal partition. We can construct a diagonal diagram D^* using D as a basic building block as shown in figure 6.1. It is easy to see that every m adjacent rows (columns) of D^* contains a subdiagram D. By theorem 2.1 we know that for the full layout W corresponding to D^* the area of any three-layer wiring W' obtained under scheme I satisfies $A(W') \ge (m/(m-1) - \varepsilon)*A(W)$, where $\varepsilon > 0$ can be arbitrary small when the dimension of D^* is large. Thus, one may find a lower bound for the worst case wiring area under scheme I by just finding a m-row by n-column diagonal diagram which has no legal partition. Clearly, the smaller the m value the larger the worst case area bound. We call the component diagonal diagram D of D^* a *strip tile*, and the diagonal diagram D^* constructed by using the D's as shown in figure 6.1 the *strip tile arrangement* A_{11}.

Theorem 6.1: There exist a four-row layout in layout class (a) (see figure 6.2), a five-row layout in layout class (b) (see figure 6.3), a six-row layout in layout class (c) (see figure 6.4) and a seven-row layout in layout class (d) (see figure 6.5) whose corresponding diagonal diagrams do not have legal partitions.

Proof: For brevity the proof is omitted. □

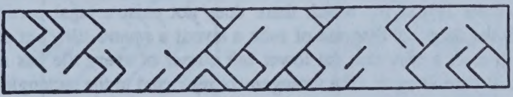


Figure 6.2

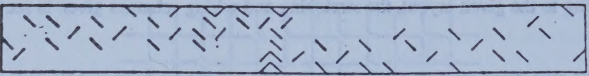


figure 6.3

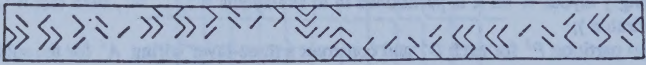


figure 6.4

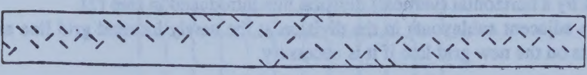


figure 6.5

Theorem 6.2: For any $\varepsilon > 0$ there exists a planar layout such that under scheme I any of its three-layer wirings W' obtained through legal partition of its corresponding diagonal diagram has area $A(W') > (c1 - \varepsilon)*A(W)$, where c1 is 4/3 if the layout is in class (a); c1 is 5/4 if the layout is in class (b); c1 is 6/5 if the layout is in class (c); c1 is 7/6 if the layout is in class (d);

Proof: The strip tile arrangement for all the layouts is A_{11}, except for the layout in class (c) which is the one given in figure 6.6. For the layout in class (c) one also needs to prove that there are no loops. For brevity the proof is omitted. □

Theorem 6.3: For any $\varepsilon > 0$ there exists a planar layout such that under scheme I any of its three-layer wirings W' has area $A(W') > (6/5 - \varepsilon)*A(W)$.

Proof: By Theorem 2.1 and Theorem 6.2. □

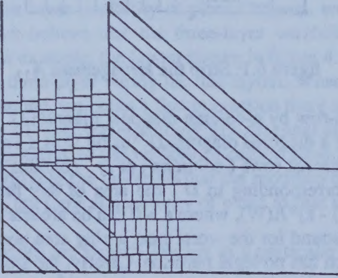


figure 6.6

VII. THREE-LAYER WIRABILITY BOUNDS FOR SCHEME II

As in the previous section, to derive lower bounds for the worst case wiring area under scheme II for each of the layout classes defined in section V, we need to find layouts with small dimensions for which legal partitions do not exist and then use these layouts to construct a larger layout with certain structure. Consider an n-row by n-column three-layer layout for which there does not exist a legal partition of its corresponding diagonal diagram. We call the diagonal diagram of such a layout a *square tile*. Let us arrange identical n-row by n-column square tiles in such a way that the lower left corner of every tile has x-coordinate i*n+j and y-coordinate j*n+i, where i and j are integers. The arrangement restricted to the rectangle formed by the horizontal lines of y-coordinate value 0 and h and vertical lines of x-coordinate values 0 and w is shown in figure 7.1. We call the arrangement restricted to this rectangle as *square tile arrangement* A_{21}. It is important to note that A_{21} is a diagonal diagram of some full planar layout.

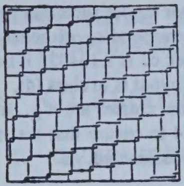


figure 7.1:

We define a window in the arrangement A_{21} as a rectangular region with boundary lines on the horizontal and vertical grid lines of G(R). We define $t(u,n) = v$, for $u > n$, if v is the smallest integer such that in A_{21} formed by unwirable rectangles of size n by n, any u row by v column window contains at least one component square tile. The computation of t(u,n) can be performed by only considering those windows whose lower left corners are located at grid points (0, y), where $0 \leq y \leq n^2$. Therefore,

$$t(u, n) = \max_y \{ \min \{ x \mid \text{the window defined by } (0, y), (0, y+u), (x, y) \text{ and } (x, y+u) \text{ in } A_{21} \text{ contains at least one n-row by n-column square tile} \} \}$$

In any (wirable) division by scheme II, if there is a sublayout with height u there must be at least $\lfloor h/u \rfloor$ horizontal division lines and there must be at least $\lfloor w/(t(u,n)-1) \rfloor$ vertical division lines. Therefore, the area is at least $(1 + 1/u)*(1 + 1/(t(u,n)-1))*A(W)$. Without loss of generality assume $w = h$ and $\lfloor w/(t(u,n)-1) \rfloor = w/(t(u,n)-1)$. Then, a lower bound for c_2 under scheme II, when obtaining a wiring through legal partition is considered, is given by

$$c_2 = \min\{ [1+1/u]*[1+1/(t(u, n)-1)] \mid n \leq u \leq n^2 \}.$$

To obtain a larger c_2 value, a smaller t(u, n) value is required; and to get a smaller t(u, n) value, smaller n value is required.

Theorem 7.1: There exist a 6-row by 6-column layout in class (a) (see figure 7.2), a 7-row by 7-column layout in class (b) (see figure 7.3), a 10-row by 10-column layout in class (c) (see figure 7.4), and a 15-row by 15-column layout in class (d) (see figure 7.5) whose corresponding diagonal diagrams do not have legal partitions.
Proof: For brevity the proof is omitted. □

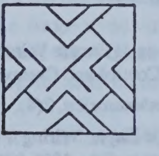


Figure 7.2

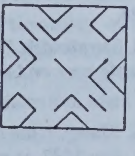


Figure 7.3

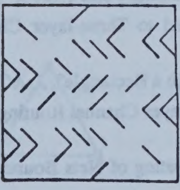


Figure 7.4

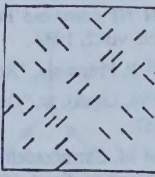


Figure 7.5

Theorem 7.2: For any $\varepsilon > 0$ there exists a planar layout such that under scheme II any of its three-layer wirings W' obtained through legal partition of its corresponding diagonal diagram has area $A(W') > (c2 - \varepsilon)*A(W)$, where c2 is 1.173333 if the layout is in class (a); c2 is 1.134375 if the layout is in class (b); c2 is 1.075630 if the layout is in class (c); and c2 is 1.060950 if the layout is in class (d).
Proof: The square tile arrangement for all layout classes are A_{11}. In the tile arrangement we use square tiles of class (c) similar to the one shown in figure 7.4 to ensure that the standard full layout corresponding to the tile arrangement is loop free. For brevity the proof is omitted. ☐

Theorem 7.3: For any $\varepsilon > 0$ there exists a planar layout such that under scheme II any of its three-layer wirings W' has area $A(W') > (1.075630 - \varepsilon)*A(W)$.
Proof: By Theorem 2.1 and Theorem 7.2, the proof is straightforward. ☐

VIII. DISCUSSIONS

We considered two different layer assignment schemes for the layer assignment problem for planar layouts. We showed the lower bounds 1.20 and 1.075630 for the worst case three-layer wiring area bounds under stretching schemes I and II, respectively. The upper bound 4/3 for the approximation factors under Scheme I by using the algorithms given in [BrS] and [GZ] is close to the lower bound 1.20 developed in this paper. Using the structure of the standard full layout W, we classified planar layouts into four different classes. We considered the problem of constructing a three-layer wiring by finding a legal partition of the diagonal diagram corresponding to the stretched layout. The lower bounds we found for the approximation factors under Scheme II are much smaller than those for the Scheme I. This provides evidence that better approximation factors can be obtained under Scheme II. Under scheme I the lower bound for the worst case wiring area bound is 4/3, when a wiring constructed by finding a legal partition of the diagonal diagram corresponding to the stretched layout is concerned. This indicates that the bound 4/3 is tight unless other wiring methods are employed. Several techniques are proposed in [GZ]. We believe that the combination of the two stretching schemes considered in this paper and some wiring techniques proposed in [GZ] may result in better wiring area bounds.

IX. ACKNOWLEDGEMENT

We wish to thank Franco P. Preparata, Donna J. Brown, Majid Sarrafzadeh and Ioannis Tollis for pointing out the errors in the earlier version of this paper.

X. REFERENCES

[BB] Brady, M. L. and D. J. Brown, "VLSI Routing: Four Layers Suffice", Advances in Computing Research, vol.2, 1984.

[BrS] Brady, M. L. and M. Sarrafzadeh, "Layout Stretching to Ensure Wirability," Proceedings of the 21st of the Johns Hopkin Conference on Communication, Control and Computing, March 1987.

[F] Frank, A., "Disjoint Paths in Rectilinear Grid", *Combinatorica*, 2(4), 1982.

[GZ] Gonzalez, T. and S. Zheng, "Stretching and Three-Layer Wiring Planar Layouts, UCSB Technical Report TRCS 87-10, Aug 1987, to appear Proceedings of the 1987 ICCD Conference.

[Le] Leighton, F. T, "New Lower Bound for Channel Routing", draft, 1981.

[Li] Lipski, W. Jr, "AN NP-complete Problem Related to Three-layer Channel Routing", Advances in Computing Research, vol.2, 1984.

[MP] Mehlhorn, K. and F. P. Preparata, "Routing Through a Rectangle", *J. ACM*, vol. 33, no. 1, 1986.

[PL] Preparata, F. P and W. Lipski, Jr, "Optimal Three-layer Channel Routing", *IEEE Trans. Comput.*, 33, 5 (May 1984), 427-437.

[PS] Preparata, F. P. and M. Sarrafzadeh, "Channel Routing of Nets Bounded Degree", *VLSI: Algorithms and Architectures*, North-Holland, 1984.

[SP] Sarrafzadeh, M. and F. P. Preparata, F. P., "Compact Channel Routing of Multiterminal Nets", Tech. Rep. ACT-44, Coordinated Science Lab., Univ. of Illinois, 1983.

ON FINDING OBSTRUCTION SETS AND POLYNOMIAL-TIME ALGORITHMS FOR GATE MATRIX LAYOUT[*]

R.L. BRYANT,[1] M.R. FELLOWS,[2] N.G. KINNERSLEY,[1] AND M.A. LANGSTON[1]

1. OVERVIEW

We consider the following NP-complete combinatorial problem.

Gate Matrix Layout (GML)

Instance: A Boolean matrix M and an integer k.

Question: Can we permute the columns of M so that, if in each row we change to $*$ every 0 lying between the row's leftmost and rightmost 1, then no column contains more than k 1s and $*$s?

This problem has been the focus of much attention, arising in a number of VLSI layout styles, including gate matrix, PLAs under multiple folding, Weinberger arrays and others.

An instance of GML may be modeled as a graph $G = (V, E)$ by a technique developed in [FL1]. A graph H is said to be a *minor* of G, written $H \leq G$, if a graph isomorphic to H can be obtained from a subgraph of G by contracting edges. For fixed k, the family $F(k)$ of graphs representing instances that can be laid out in k tracks is *closed* under the minor ordering and excludes a planar graph. Thus it follows that determining if a graph G is a member of $F(k)$ can be accomplished (in principle) in quadratic time [FL2, FL3, RS].

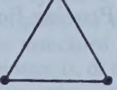
The quadratic-time algorithm proven to exist is based on the fact that the complement of $F(k)$ contains a *finite* number of minimal elements in the minor ordering. We shall refer to this minor-minimal set as the "obstruction set" for $F(k)$. Because the family $F(k)$ is closed under minors, a graph G is in $F(k)$ if and only if there is no H in the obstruction set for $F(k)$ with $H \leq G$. This result is *nonconstructive* and begs three important questions:

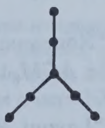
1) What are the minor-minimal obstructions for fixed values of k?
2) Can we use the decision algorithm to build an optimal layout?
3) Can we avoid the huge constants of proportionality of [RS]?

We herein focus on question 1). Question 2) has been answered in the affirmative [BFL]. Question 3) is an open and interesting topic for future study.

2. A SAMPLING OF RESULTS

Theorem 1. The obstruction set for $F(2)$ contains only the two graphs depicted below.





---

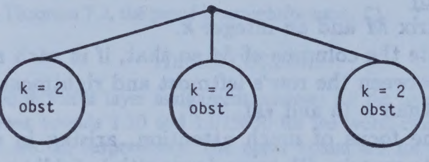
[1] Department of Computer Science, Washington State University, Pullman, WA 99164–1210.

[2] Department of Computer Science, University of Idaho, Moscow, ID 83843.

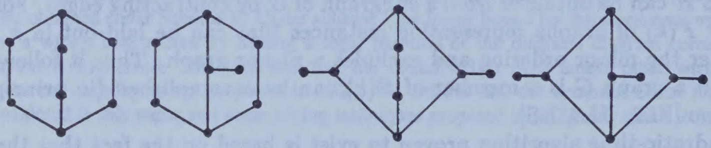
[*] This research is supported in part by the Washington State Technology Center and the National Science Foundation under grants ECS–8403859 and MIP–8603879.

Now consider $F(3)$. Since K_4 is an obstruction, any yes-instance must be series-parallel (and planar). This can be checked in linear time. Also, if $|E| > 2|V| - 3$, then necessarily $G \geq K_4$. A practical algorithm makes this easy test as a first step.

Theorem 2. If the two obstructions for $F(2)$ are connected in the manner shown below, twenty distinct obstructions for $F(3)$ are obtained: ten trees, six graphs containing a single triangle, three containing two triangles and one with three triangles. Moreover, the ten trees so obtained are the only tree obstructions for $F(3)$.



Theorem 3. K_4 and the graphs depicted below are the only members of the obstruction set for $F(3)$ that are not outerplanar.



Theorem 4. No obstruction for $F(3)$ has more than four internal faces.

The last result allows us to apply a series of combinatorial arguments to bound the search space for all obstructions for $F(3)$. We have isolated 110 obstructions as of this writing and conjecture that there are no more.

REFERENCES

[BFL] D. Brown, M. Fellows and M. Langston, "Nonconstructive Polynomial-Time Decidability and Self-Reducibility," *Princeton Forum on Algorithms and Complexity*, 1987.

[FL1] M. Fellows and M. Langston, "Nonconstructive Advances in Polynomial-Time Complexity," *Info. Proc. Letters*, to appear.

[FL2] ―――――, "Nonconstructive Tools for Proving Polynomial-Time Decidability," *J. of the ACM*, to appear.

[FL3] ―――――, "Layout Permutation Problems and Well-Partially-Ordered Sets," to appear.

[RS] N. Robertson and P. Seymour, "Disjoint Paths—A Survey," *SIAM J. Alg. Disc. Meth.* 6 (1985), 300–305.

Optimal Joint Parameter and State Estimation of Linear Stochastic MIMO Systems Using Pseudo-Linear Identification (PLID)

by

Mark A. Hopkins and Hugh F. VanLandingham
Virginia Polytechnic Institute and State University (Virginia Tech)

This paper presents results on a new method of joint parameter and state estimation, called pseudo-linear identification (PLID), applied to linear, stochastic, multiple-input, multiple-output (MIMO), completely observable, completely controllable systems. These results extend the work of Salut, et.al. [1], *to the more general (and more practical) case where there is noise corrupting the input signal and the output measurements. It is proved that PLID is the optimal state and parameter estimator, under standard gaussian assumptions, for* time-invariant *systems of the class described above.*

In order to use PLID, the *structure* (*i.e.*, the system order and the various observability indexes) of the unknown system must be known (or assumed) in advance. It is also assumed that the system is strictly proper (*i.e.*, there is positive delay from each input to every output). No assumptions are required concerning pole and zero placement, with the exception that no pole/zero cancellations may be present in the system, unless they have been deducted from the system order.

The standard approach to joint estimation of parameters and states, which is an explicitly non-linear problem, uses the Extended Kalman Filter (EKF) to linearize about the current estimate. PLID is derived in a way that makes the nonlinearities *implicit*, yielding an explicitly linear algorithm (hence the name). The proof of its optimality for time-invariant systems (and proof of convergence, to be presented elsewhere) is possible as a result of this linearity.

The PLID algorithm is developed for time-varying systems with white noise parameter variations. No results have been established for the time-varying case, which is quite difficult from the theoretical point of view. A typical approach is to assume the parameter variations are slow enough that, for practical purposes, they can be assumed constant. Then the results established for time-invariant systems are invoked, although, strictly speaking, they do not hold in such a case.

§1 develops the extended state representation, from which §2 derives the essential form of the PLID algorithm. §3 proves the time-optimal convergence of the deterministic algorithm, and §4 presents the complete stochastic PLID algorithm. §5 proves that PLID is the optimal estimator, under standard gaussian assumptions. §6 presents a comparison of PLID and the EKF.

1. Extended State Representation of MIMO Systems

This section extends [2] to stochastic multiple-input multiple-output (MIMO) systems, resulting in the extended state model for the general observable MIMO system. The extended state representation generalizes [1] to the case where there is input and measurement (output) noise.

Suppose an unknown system S is MIMO, linear, time-invariant, completely controllable, and completely observable. Let p denote the number of system outputs, $y_1(k), \ldots, y_p(k)$, and let $n_1, \ldots, n_p$ be the observability indices associated with those outputs, n_i being associated with $y_i(k)$. The total number n of states in the system is, of course, the sum of the observability indices. Also, let m denote the number of system inputs, $u_1(k), \ldots, u_m(k)$.

There is no loss of generality in assuming a strictly proper system, since every real system has non-zero delay from input to output. So, by the assumption of complete observability and controllability, there is a transfer function for the system in the form:

$$\left[\begin{array}{c|c|c} z^{n_1} - a_{1,1}^{n_1-1}\, z^{n_1-1} - \dots - a_{1,1}^{0} & \dots & -\, a_{p,1}^{n_1-1}\, z^{n_1-1} - \dots - a_{p,1}^{0} \\ \hline \vdots & & \vdots \\ \hline -\, a_{1,p}^{n_1-1}\, z^{n_p-1} - \dots - a_{1,p}^{0} & \dots & z^{n_p} - a_{p,p}^{n_p-1}\, z^{n_p-1} - \dots - a_{p,p}^{0} \end{array}\right] \begin{bmatrix} Y_1(z) \\ \vdots \\ Y_p(z) \end{bmatrix} = \left[\begin{array}{c|c|c} b_{1,1}^{n_1-1}\, z^{n_1-1} + \dots + b_{1,1}^{0} & \dots & b_{m,1}^{n_1-1}\, z^{n_1-1} + \dots + b_{m,1}^{0} \\ \hline \vdots & & \vdots \\ \hline b_{1,p}^{n_p-1}\, z^{n_p-1} + \dots + b_{1,p}^{0} & \dots & b_{m,p}^{n_p-1}\, z^{n_p-1} + \dots + b_{m,p}^{0} \end{array}\right] \begin{bmatrix} U_1(z) \\ \vdots \\ U_m(z) \end{bmatrix}. \tag{1.1}$$

The multivariable observable-canonic state equation, following the outline in Chen [3], is given by Equations 1.2 and 1.3. Since we want to develop the stochastic equations, input vector u_k and output vector z_k are assumed to have additive noises v_k and w_k, respectively, and the states are assumed to have noise vector $\Xi_x(k)$ added directly. Hence,

$$x_{k+1} = \begin{bmatrix} x_{1,1}(k+1) \\ \vdots \\ x_{1,n_1}(k+1) \\ \vdots \\ x_{p,1}(k+1) \\ \vdots \\ x_{p,n_p}(k+1) \end{bmatrix} = \left[\begin{array}{c|c|c} 0 \dots 0\ a_{1,1}^{0} & \dots & 0 \dots 0\ a_{p,1}^{0} \\ 1 \dots 0\ a_{1,1}^{1} & \dots & 0 \dots 0\ a_{p,1}^{1} \\ \cdot & & \vdots \quad \vdots \\ 0 \dots 1 a_{1,1}^{n_1-1} & \dots & 0 \dots 0 a_{p,1}^{n_1-1} \\ \hline \vdots & & \vdots \\ \hline 0 \dots 0\ a_{1,p}^{0} & \dots & 0 \dots 0\ a_{p,p}^{0} \\ 0 \dots 0\ a_{1,p}^{1} & \dots & 1 \dots 0\ a_{p,p}^{1} \\ \vdots \quad \vdots \quad \vdots & & \cdot \\ 0 \dots 0\ a_{1,p}^{n_p-1} & \dots & 0 \dots 1\ a_{p,p}^{n_p-1} \end{array}\right] \begin{bmatrix} x_{1,1}(k) \\ \vdots \\ x_{1,n_1}(k) \\ \vdots \\ x_{p,1}(k) \\ \vdots \\ x_{p,n_p}(k) \end{bmatrix} + \begin{bmatrix} \xi_{1,1}(k) \\ \vdots \\ \xi_{1,n_1}(k) \\ \vdots \\ \xi_{p,1}(k) \\ \vdots \\ \xi_{p,n_p}(k) \end{bmatrix} \tag{1.2}$$

$$+ \left[\begin{array}{c|c|c} b_{1,1}^{0} & \dots & b_{m,1}^{0} \\ \vdots & & \vdots \\ b_{1,1}^{n_1-1} & \dots & b_{m,1}^{n_1-1} \\ \hline \vdots & & \vdots \\ \hline b_{1,p}^{0} & \dots & b_{m,p}^{0} \\ \vdots & & \vdots \\ b_{1,p}^{n_p-1} & \dots & b_{m,p}^{n_p-1} \end{array}\right] \begin{bmatrix} u_1(k) + v_1(k) \\ \vdots \\ u_m(k) + v_m(k) \end{bmatrix} \stackrel{\Delta}{=} A\, x_k + \Xi_x(k) + B\,(u_k + v_k).$$

$$z_k = \begin{bmatrix} z_1(k) \\ \vdots \\ z_p(k) \end{bmatrix} = \left[\begin{array}{c|c|c} 0 \dots 0\ 1 & \dots & 0 \dots 0\ 0 \\ \hline \vdots & \vdots & \vdots \\ \hline 0 \dots 0\ 0 & \dots & 0 \dots 0\ 1 \end{array}\right] \begin{bmatrix} x_{1,1}(k) \\ \vdots \\ x_{1,n_1}(k) \\ \vdots \\ x_{p,1}(k) \\ \vdots \\ x_{p,n_p}(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ \vdots \\ w_m(k) \end{bmatrix} \stackrel{\Delta}{=} C\, x_k + w_k\,. \tag{1.3}$$

The general form of the rows of Equation 1.2 is:

$$x_{i,\,j}(k+1) = \Big[x_{i,\,j-1}(k) + a_{1,i}^{j-1}\, x_{1,n_1}(k) + \cdots + a_{p,i}^{j-1}\, x_{p,n_p}(k) + \xi_{i,j}(k) \\ + b_{1,i}^{j-1}\,[u_1(k) + v_1(k)] + \cdots + b_{m,i}^{j-1}\,[u_m(k) + v_m(k)]\Big]. \tag{1.4}$$

Substituting the general row $z_i(k) = x_{i,n_i}(k) + w_i(k)$ of Equation 1.3. into Equation 1.4,

$$x_{i,\,j}(k+1) = \Big[x_{i,\,j-1}(k) + a_{1,i}^{j-1}\,[z_1(k) - w_1(k)\,] + \cdots + a_{p,i}^{j-1}\,[z_p(k) - w_p(k)\,] \\ + \xi_{i,j}(k) + b_{1,i}^{j-1}\,[u_1(k) + v_1(k)\,] + \cdots + b_{m,i}^{j-1}\,[u_m(k) + v_m(k)\,]\Big]. \tag{1.5}$$

Next, the *extended state vector* is defined as

$$s_k \overset{\Delta}{=} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} \overset{\Delta}{=} \begin{bmatrix} x_k \\ \theta_A^1 \\ \vdots \\ \theta_A^p \\ \theta_B^1 \\ \vdots \\ \theta_B^m \end{bmatrix}, \text{ where } \theta_A^i \overset{\Delta}{=} \begin{bmatrix} a_{i,1}^0 \\ \vdots \\ a_{i,1}^{n_1-1} \\ \vdots \\ a_{i,p}^0 \\ \vdots \\ a_{i,p}^{n_p-1} \end{bmatrix},\ \theta_B^i \overset{\Delta}{=} \begin{bmatrix} b_{i,1}^0 \\ \vdots \\ b_{i,1}^{n_1-1} \\ \vdots \\ b_{i,p}^0 \\ \vdots \\ b_{i,p}^{n_p-1} \end{bmatrix}. \tag{1.6}$$

Note that θ_A^i is the $(n_1 + \cdots + n_i)^{th}$-column of system matrix A; and θ_B^i is the i^{th}-column of input matrix B. The elements of parameter vectors θ_A^i and θ_B^i are assumed to be states in the extended state vector. Therefore, to make the model completely general, they are assumed to have additive noise, represented by the elements of noise vectors Ξ_A^i and Ξ_B^i (for θ_A^i and θ_B^i, respectively).

With Equation 1.6 defining the extended state vector s_k, and with the general form identified in Equation 1.5, the *extended system* $\overline{S}$ can be written as

$$\begin{aligned} s_{k+1} &= F_k\, s_k + G_k\, \eta_k \\ z_k &= H_k\, s_k + w_k \end{aligned} \tag{1.7}$$

where
$$F_k = \left[\begin{array}{c|c|c|c|c|c|c|c|c} J_{n_1} & \cdots & 0 & & & & & & \\ & & & z_1(k)\, I_n & \cdots & z_p(k)\, I_n & u_1(k)\, I_n & \cdots & u_m(k)\, I_n \\ 0 & \cdots & J_{n_p} & & & & & & \\ \hdashline \multicolumn{3}{c|}{0_{(m+p)n \times n}} & \multicolumn{6}{c}{I_{(m+p)n}} \end{array}\right]$$

and
$$H_k = \left[\begin{array}{c|c|c|c} 0 \ldots 0\ 1 & \cdots & 0 \ldots 0\ 0 & \\ \vdots & & \vdots & 0_{p \times (m+p)n} \\ 0 \ldots 0\ 0 & \cdots & 0 \ldots 0\ 1 & \end{array}\right].$$

Note that in matrix F_k, J_{n_i} is an $n_i \times n_i$ lower Jordan block of zero eigenvalues. Also note that there are $(m+p+1)n$ states in the extended system $\overline{S}$.

Defining the extended system input noise vector

$$\eta_k = \begin{bmatrix} w_k \\ v_k \\ \Xi_k \end{bmatrix} \overset{\Delta}{=} \begin{bmatrix} w_k \\ v_k \\ \Xi_x(k) \\ \Xi_A^1(k) \\ \vdots \\ \Xi_A^p(k) \\ \Xi_B^1(k) \\ \vdots \\ \Xi_B^m(k) \end{bmatrix}, \text{ then } G_k = \left[\begin{array}{c|c|c|c|c|c|c} -\theta_A^1 & & -\theta_A^p & \theta_B^1 & & \theta_B^m & \\ 0 & \cdots & 0 & 0 & \cdots & 0 & I_{(m+p+1)n} \\ & & & & & & \end{array}\right]. \tag{1.8}$$

The noise vectors v_k, w_k, Ξ_k are assumed to be mutually independent, zero mean, gaussian white noise vectors. It is assumed that the respective covariances are known for all time k; they will be denoted

$$E[v_k v_k^T] = Q_k, \quad E[w_k w_k^T] = R_k, \quad \text{and} \quad E[\Xi_k \Xi_k^T] = \Sigma_k = block\ diag\,(\Sigma_k^{xx}, \Sigma_k^{\theta\theta}). \tag{1.9}$$

Hence, $E[\eta_k \eta_k^T] = block\ diag\,(R_k, Q_k, \Sigma_k) \overset{\Delta}{=} Q_0(k)$, and $E[\eta_k w_k^T] = [R_k \mid 0 \mid 0]^T \overset{\Delta}{=} S_k$. (1.10)

2. Derivation of the Algorithm

We want to compute a conditional minimum-mean-square-error recursive prediction $\hat{s}_{k+1|k}$ of the extended state vector s_{k+1} of $\bar{S}$ (Equation 1.7) using the linear form $\hat{s}_{k+1|k} = M_k \hat{s}_{k|k-1} + \bar{K}_k z_k$. Note $\bar{K}_k$ is a gain matrix chosen to minimize the conditional mean square error of the prediction. The prediction is to be conditioned on the increasing sequence of sub-σ-fields ψ_k generated by the increasing sequence of sets of data $\{ z_0, \dots, z_k \}$. Thus $\hat{s}_{k+1|k} = E(s_{k+1} \mid \psi_k)$ (also denoted $E_{\psi_k}(s_{k+1})$).

The predictor is required to be unbiased; *i.e.*, the expectations

$$\begin{aligned} E[s_{k+1} \mid \psi_k] &= E[\hat{s}_{k+1|k} \mid \psi_k] = E[M_k \hat{s}_{k|k-1} + \bar{K}_k z_k \mid \psi_k] \\ F_k s_k &= M_k E(s_k \mid \psi_k) + \bar{K}_k H_k E(s_k \mid \psi_k) = M_k s_k + \bar{K}_k H_k s_k. \end{aligned} \tag{2.1}$$

$$\text{Therefore,} \quad F_k = M_k + \bar{K}_k H_k \quad \rightarrow \quad \hat{s}_{k+1|k} = (F_k - \bar{K}_k H_k)\, \hat{s}_{k|k-1} + \bar{K}_k z_k. \tag{2.2}$$

Now define the prediction error, recalling that the input noise vector is $\eta_k = [w_k^T \mid v_k^T \mid \Xi_k^T]^T$,

$$e_{k+1|k} \overset{\Delta}{=} \hat{s}_{k+1|k} - s_{k+1} = (F_k - \bar{K}_k H_k)\, e_{k|k-1} + \bar{K}_k w_k - G_k \eta_k \tag{2.3}$$

Define the error covariance, again conditioning on the increasing sequence of sub-σ-fields ψ_k, by

$$\begin{aligned} P_{k+1|k} &\overset{\Delta}{=} E\left[e_{k+1|k}\, e_{k+1|k}^T \mid \psi_k \right] \\ &= E\Big[(F_k - \bar{K}_k H_k)\, e_{k|k-1}\, e_{k|k-1}^T (F_k - \bar{K}_k H_k)^T + K_{k|k}\, w_k w_k^T \bar{K}_k^T \\ &\quad + G_k \eta_k \eta_k^T G_k^T - \bar{K}_k w_k \eta_k^T G_k^T - G_k \eta_k w_k^T \bar{K}_k^T \\ &\quad - (F_k - \bar{K}_k H_k)\, e_{k|k-1} \eta_k^T G_k^T - G_k \eta_k e_{k|k-1}^T (F_k - \bar{K}_k H_k)^T \\ &\quad + (F_k - \bar{K}_k H_k)\, e_{k|k-1} w_k^T \bar{K}_k^T + \bar{K}_k w_k e_{k|k-1}^T (F_k - \bar{K}_k H_k)^T \mid \psi_k \Big]. \end{aligned} \tag{2.4}$$

Taking the expectations, the last four terms on the right side are zero, and

$$\begin{aligned} P_{k+1|k} = \Big[&(F_k - \bar{K}_k H_k)\, P_{k|k-1} (F_k - \bar{K}_k H_k)^T + \bar{K}_k R_k \bar{K}_k^T + E(G_k Q_0(k) G_k^T \mid \psi_k) \\ &- \bar{K}_k S_k^T E(G_k^T \mid \psi_k) - E(G_k \mid \psi_k) S_k \bar{K}_k^T \Big]. \end{aligned} \tag{2.5}$$

The gain $\bar{K}_k$ is to be computed so as to minimize the mean-square error of the prediction $\hat{s}_{k+1|k}$, so choose the functional $L_{k+1|k} = E[\, e_{k+1|k}^T e_{k+1|k} \mid \psi_k] = tr\,(P_{k+1|k})$.

The minimizing gain $K_{k|k}$ is found by solving $\left[\frac{\partial}{\partial \overline{K}_k} L_{k+1|k} \right]_{\overline{K}_k = K_{k|k}} = 0.$ (2.6)

$E_{\psi_k}(\ )$ will also indicate conditional mean relative to ψ_k. The partial derivative is

$$0 = \frac{\partial}{\partial \overline{K}_k} tr\{[\overline{K}_k (H_k P_{k|k-1} H_k^T + R_k)\overline{K}_k^T] - 2[(F_k P_{k|k-1} H_k^T + E_{\psi_k}[G_k]S_k)\overline{K}_k^T]\}\Big|_{\overline{K}_k = K_{k|k}}$$

Hence, $K_{k|k} = [F_k P_{k|k-1} H_k^T + E_{\psi_k}(G_k) S_k](H_k P_{k|k-1} H_k^T + R_k)^{-1}.$ (2.7)

Now back-substitution of Equation 2.7 into 2.5 yields:

$$P_{k+1|k} = F_k P_{k|k-1} F_k^T + E_{\psi_k}(G_k Q_0(k) G_k^T) - K_{k|k}(H_k P_{k|k-1} H_k^T + R_k)K_{k|k}^T. \tag{2.8}$$

Equations 2.2. (with $\overline{K}_k = K_{k|k}$), 2.7 and 2.8 give the essential form of the PLID algorithm. Note the differences from the Kalman filter. In the stochastic case, more work is required to compute the two conditional expectations, as discussed in Section 4.

3. Deadbeat Convergence of Deterministic PLID

The deterministic PLID algorithm is given by Equations 2.2 (with $\overline{K}_k = K_{k|k}$), 2.7, and and 2.8, with all the noise covariances and cross-covariances set to zero, and the inverse replaced by the pseudo-inverse. Thus, by definition, the unknown parameters must be time-invariant. This section presents proof that the deterministic PLID algorithm converges to the true values of the states and parameters, and converges in the minimum possible time (hence is deadbeat). The proof is based upon demonstrating the connection between the observability matrix of the extended system and the error propagation, leading naturally to a persistency of excitation requirement.

Let us start by addressing some issues of notation. Denoting the rows of the output matrix by

$H_k = \begin{bmatrix} h_0(k) \\ \vdots \\ h_{p-1}(k) \end{bmatrix}$, the observability matrix $\Phi_k = \begin{bmatrix} H_0 \\ H_1 F_0 \\ \vdots \\ H_k F_{k-1} \dots F_0 \end{bmatrix}$ will be decomposed into rows

called the observation vectors $\varphi_{kp+j} \overset{\Delta}{=} h_j(k) F_{k-1} \dots F_0, \quad j \in \{0, \dots, p-1\}.$ (3.1)

Lemma 3.1: Suppose $P_{0|-1}$ is symmetric positive definite.
For all $k \geq 0$, and $j \in \{0, \dots, p-1\}$, $P_{k|k-1} h_j^T(k) \neq 0$ iff $h_j(k) P_{k|k-1} h_j^T(k) \neq 0$.

Proof: " → " Suppose $P_{k|k-1} h_j^T(k) \neq 0$. Let $m = rank(P_{k|k-1}) > 0$. Since $P_{k|k-1}$ is symmetric non-negative definite, it can be written as a sum of m vector outer products. That is,

$$0 \neq P_{k|k-1} h_j^T(k) = \left[\sum_{i=1}^{m} c_i(k) c_i^T(k) \right] h_j^T(k).$$

Therefore, for some $i \in \{1, \dots, m\}$, $0 \neq c_i^T(k) h_j^T(k)$. It follows that

$$0 \neq \sum_{i=1}^{m} [h_j(k) c_i(k)][c_i^T(k) h_j^T(k)] = h_j(k) \left[\sum_{i=1}^{m} c_i(k) c_i^T(k) \right] h_j^T(k) = h_j(k) P_{k|k-1} h_j^T(k).$$

" ← " (by contradiction) Suppose $h_j(k) P_{k|k-1} h_j^T(k) \neq 0$. Assume that $P_{k|k-1} h_j^T(k) = 0$.
Then $h_j(k) [P_{k|k-1} h_j^T(k)] = 0$, which is a contradiction. ■

Lemma 3.2: Suppose $P_{0|-1}$ is symmetric positive definite. For all $k \geq 0$, and $j \in \{0, \dots, p-1\}$, $P_{k|k-1} h_j^T(k)$ is in the null space of $(F_k - K_{k|k} H_k)$.

Proof: The proof uses the full output matrix H_k. "#" denotes the pseudo-inverse ($R_k = 0$).
Case 1: $H_k P_{k|k-1} H_k^T = 0$. By Lemma 3.1, $P_{k|k-1} H_k^T = 0$, hence $(F_k - K_{k|k} H_k) P_{k|k-1} H_k^T = 0$.
Case 2: $H_k P_{k|k-1} H_k^T \neq 0$. Dropping the time notation from $h_i(k)$ for compactness,

$$\begin{aligned}(F_k - K_{k|k} H_k) P_{k|k-1} H_k^T &= \left[F_k - F_k P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T]^{\#} H_k\right] P_{k|k-1} H_k^T \\ &= F_k P_{k|k-1} H_k^T \left[I - [H_k P_{k|k-1} H_k^T]^{\#} [H_k P_{k|k-1} H_k^T]\right] \\ &= F_k \left[P_{k|k-1} h_0^T \mid \dots \mid P_{k|k-1} h_{p-1}^T\right] \begin{bmatrix} 1 - (h_0 P_{k|k-1} h_0^T)^{\#} (h_0 P_{k|k-1} h_0^T) & & * \\ & \ddots & \\ * & & 1 - (h_{p-1} P_{k|k-1} h_{p-1}^T)^{\#} (h_{p-1} P_{k|k-1} h_{p-1}^T) \end{bmatrix} \\ &= F_k \times (0) = 0\end{aligned}$$

with the last line due to Lemma 3.1, because

$$P_{k|k-1} h_i^T(k) \neq 0 \quad \rightarrow \quad [1 - (h_i(k) P_{k|k-1} h_i^T(k))^{\#} h_i(k) P_{k|k-1} h_i^T(k)] = 0,$$

and the fact that $(H_k P_{k|k-1} H_k^T)^{\#} H_k P_{k|k-1} H_k^T$ is symmetric nonnegative definite.
Since the lemma holds for H_k, then it holds for each row, $h_j(k)$. ■

Theorem 3.3: Suppose $P_{0|-1}$ is symmetric positive definite.
The observation vectors $\{\varphi_0^T, \dots, \varphi_{kp+p-1}^T\}$ are in the null space of

$$(F_k - K_{k|k} H_k)(F_{k-1} - K_{k-1|k-1} H_{k-1}) \dots (F_0 - K_{0|0} H_0) P_{0|-1}.$$

Proof: By Lemma 3.2, for each $j \in \{0, \dots, p-1\}$, $0 = (F_k - K_{k|k} H_k) P_{k|k-1} h_j^T(k)$.
Now, for the deterministic case, $\eta_k \equiv 0$, and $w_k \equiv 0$. So the error covariance matrix, $P_{k|k-1} = (F_{k-1} - K_{k-1|k-1} H_{k-1}) P_{k-1|k-2} F_{k-1}^T$, can be expanded by self-iteration; substituting the expansion into the equation from Lemma 3.2, then applying Equation 3.1,

$$\begin{aligned} 0 &= (F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1} F_0^T F_1^T \dots F_{k-1}^T h_j^T(k) \\ &= (F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1} \varphi_{pk+j}^T \quad \text{for all } j = 0, \dots, p-1 \end{aligned} \tag{3.2}$$

Generalizing Equation 3.2, for any $i = 0, \dots, k$,

$$0 = (F_i - K_{i|i} H_i) \dots (F_0 - K_{0|0} H_0) P_{0|-1} \varphi_{p\,i+j} \tag{3.3}$$

Hence (premultiplying appropriately), for all $i = 0, \dots, k$, and all $j = 0, \dots, p-1$,

$$0 = (F_k - K_{k|k} H_k) \dots \left[(F_i - K_{i|i} H_i) \dots (F_0 - K_{0|0} H_0) P_{0|-1} \varphi_{p\,i+j}\right] \qquad ■$$

Theorem 3.3 implies $nullity\left[(F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1}\right] \geq rank[\Phi_k]$.
The next theorem proves that the equality always holds.

Theorem 3.4: Suppose $P_{0|-1}$ is symmetric positive definite.
Let $N = (m + p + 1)n$, the number of states in the extended state vector s_k.
Let A be the *ordered set* of indices α_i, $i \in \{0, \dots, N-1\}$, created by including each index $kp+j$ for which $k \geq 0$ and $j \in \{0, \dots, p-1\}$ satisfy $h_j(k) P_{k|k-1} h_j^T(k) \neq 0$.

(1) The set of observation vectors $\{\varphi_{\alpha_0}, \dots, \varphi_{\alpha_{N-1}}\}$ is linearly independent.
(2) If $rank(\Phi_k) = m$, then the range space of $(F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1}$ is spanned by $\{\varphi_{\alpha_m}^T \dots \varphi_{\alpha_{N-1}}^T\}$.

Proof: In this proof, let k_i be such that $p k_i \leq \alpha_i < p(k_i + 1)$.
As a result, by the construction of the set A, $h_{\alpha_i - pk_i}(k_i) P_{k_i|k_i-1} h_{\alpha_i - pk_i}^T(k_i) \neq 0$.

Part (1) (by contradiction): Suppose that, for some $i \in \{1, \dots, N-1\}$, $\varphi_{\alpha_i} = \sum_{j=0}^{i-1} \beta_j \varphi_{\alpha_j}$.

By Theorem 3.3, $0 = (F_{k_{i-1}-1} - K_{k_{i-1}-1|k_{i-1}-1} H_{k_{i-1}-1}) \dots (F_0 - K_{0|0} H_0) P_{0|-1} \varphi_{\alpha_i}^T$.
So, premultiplying appropriately, and expanding $\varphi_{\alpha_i}^T$ using Equation 3.1,

$$\begin{aligned} 0 &= (F_{k_i-1} - K_{k_i-1|k_i-1} H_{k_i-1}) \dots (F_0 - K_{0|0} H_0) P_{0|-1} F_0^T \dots F_{k_i-1}^T h_{\alpha_i - pk_i}^T(k_i) \\ &= P_{k_i|k_i-1} h_{\alpha_i - pk_i}^T(k_i). \end{aligned} \tag{3.4}$$

Therefore, $0 = h_{\alpha_i - pk_i}(k_i) P_{k_i|k_i-1} h_{\alpha_i - pk_i}^T(k_i) \rightarrow \alpha_i \notin A$, contradicting a premise of the theorem.
Part (2) (also by contradiction): Suppose that, for some $d \in \{1, \dots, N\text{-}1\text{-}i\}$ it is true that

$$0 = (F_{k_i-1} - K_{k_i-1|k_i-1} H_{k_i-1}) \dots (F_0 - K_{0|0} H_0) P_{0|-1} \varphi_{\alpha_{i+d}}^T. \tag{3.5}$$

Then, premultiplying appropriately, and expanding $\varphi^T_{\alpha_{i+d}}$ using Equation 3.1,

$$\begin{aligned} 0 &= (F_{k_{i+d}-1} - K_{k_{i+d}-1|k_{i+d}-1} H_{k_{i+d}-1}) \dots (F_0 - K_{0|0} H_0) P_{0|-1} F_0^T \dots \\ &\qquad \dots F^T_{k_{i+d}-1} h^T_{\alpha_{i+d}-pk_{i+d}}(k_i+d) \qquad (3.6) \\ &= P_{k_{i+d}|k_{i+d}-1} h^T_{\alpha_{i+d}-pk_{i+d}}(k_i+d) . \end{aligned}$$

Hence, $0 = h_{\alpha_{i+d}-pk_{i+d}}(k_i+d) \; P_{k_{i+d}|k_{i+d}-1} \; h^T_{\alpha_{i+d}-pk_{i+d}}(k_i+d) \rightarrow \varphi_{\alpha_{i+d}} \notin A$ (contradiction). ■

Together, Theorems 3.3 and 3.4 imply that

$$\mathit{nullity}\left[(F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1}\right] = \mathit{rank}\,[\Phi_k]. \qquad (3.7)$$

The next theorem examines the consequences of Equation 3.7 for the prediction error $e_{k+1|k}$.

Theorem 3.5: Suppose $P_{0|-1}$ is symmetric positive definite.

If at time k the observability matrix has rank $(m+p+1)n$, then the prediction error $e_{k+1|k}$ is zero.

Proof: From Equation 2.3, with no noise terms, applying Theorems 3.3 and 3.4,

$$\begin{aligned} e_{k+1|k} &= (F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) e_{0|-1} \\ &= (F_k - K_{k|k} H_k) \dots (F_0 - K_{0|0} H_0) P_{0|-1} (P_{0|-1})^{-1} e_{0|-1} = 0. \end{aligned}$$ ■

The extended state cannot be uniquely identified before the generalized inverse of the observability matrix Φ_k exists (i.e., not before $\mathit{rank}(\Phi_k) = (m+p+1)n$). Therefore, the PLID algorithm is time-optimal (deadbeat), because it is precisely when the observability matrix attains rank equal to $(m+p+1)n$ that the error vector $e_{k|k-1}$ becomes zero. The input is persistently exciting if it causes the observability matrix to reach full rank.

A simple way to determine when the prediction error is zero is to count the number of times that diagonal elements of $H_k P_{k|k-1} H_k^T$ are non-zero. (The diagonal elements are $h_i(k) P_{k|k-1} h_i^T(k)$.) When the count reaches $(m+p+1)n$, the prediction error will be zero. One implication of Theorems 3.3 and 3.4 is that the PLID algorithm will converge even if there is some delay before the input becomes sufficiently exciting; this is borne out by simulations.

4. The Stochastic PLID Algorithm

Equations 2.2, 2.7, and 2.8 are the defining equations of the stochastic PLID algorithm. However, $E(G_k \mid \psi_k) = E_{\psi_k}(G_k)$ and $E(G_k Q_0(k) G_k^T \mid \psi_k) = E_{\psi_k}(G_k Q_0(k) G_k^T)$ in those equations have not yet been specified. This section specifies them, completing the stochastic PLID algorithm.

To develop the complete algorithm, decompose the various matrices according to Equation 1.6:

$$F_k = \left[\begin{array}{c|c} \multicolumn{2}{c}{F_k^x} \\ \hdashline 0 & I_{(m+p)n} \end{array}\right], \quad \Sigma_k = \begin{bmatrix} \Sigma_k^{xx} & 0 \\ 0 & \Sigma_k^{\theta\theta} \end{bmatrix}, \quad P_{k+1|k} = \begin{bmatrix} P^{xx}_{k+1|k} & P^{x\theta}_{k+1|k} \\ P^{\theta x}_{k+1|k} & P^{\theta\theta}_{k+1|k} \end{bmatrix}, \qquad (4.1)$$

$$E_{\psi_k}[G_k] = \left[\begin{array}{c|c} \multicolumn{2}{c}{E_{\psi_k}(G_k^x)} \\ \hdashline 0 & I_{(m+p)n} \end{array}\right], \text{ and } E_{\psi_k}[G_k Q_0(k) G_k^T] = \left[\begin{array}{c|c} \begin{array}{l} \Sigma_k^{xx} + \sum\limits_{i=1}^{p} \sum\limits_{j=1}^{p} r_{i,j} E_{\psi_k}[\theta_A^i (\theta_A^j)^T] \\ \quad + \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{m} q_{i,j} E_{\psi_k}[\theta_B^i (\theta_B^j)^T] \end{array} & 0 \\ \hdashline 0 & \Sigma_k^{\theta\theta} \end{array}\right].$$

where the $q_{i,j}(k)$ are the elements of Q_k , the $r_{i,j}(k)$ are the elements of R_k . It is crucial to the stochastic PLID algorithm that G_k and $G_k Q_0(k) G_k^T$ are independent of the σ-field ψ_k in all subpartitions related to the parameter estimates. That allows $K^{\theta}_{k|k}$ and $\hat{\theta}_{k+1|k}$ to be computed first, *then* the $E[G_k^x \mid \psi_k]$, which is a linear function of $\hat{\theta}_{k+1|k}$ (under standard gaussian assumptions -- see § 5). With that expectation known, $K^x_{k|k}$ can be computed, followed by $x_{k+1|k}$.

Now, with the entire gain matrix $K_{k|k}$ known, a similar argument shows that $P^{\theta\theta}_{k+1|k}$, $P^{\theta x}_{k+1|k}$, and $P^{x\theta}_{k+1|k}$ can be computed first, followed by the unknown part of $E\,(G_k\,Q_0(k)\,G_k^T \mid \psi_k)$, and finally $P^{xx}_{k+1|k}$. (Note that $E_{\psi_k}(\theta\,\theta^T) = P^{\theta\theta}_{k+1|k} + \hat{\theta}_{k+1|k}\,\hat{\theta}^T_{k+1|k}$ -- see §5.)

Thus, the stochastic PLID algorithm is given by Equations 4.2 through 4.11:

$$\tilde{K}_k = \begin{bmatrix} \tilde{K}^x_k \\ \tilde{K}^\theta_k \end{bmatrix} = F_k\,P_{k|k-1}\,H_k^T\,(\,H_k\,P_{k|k-1}\,H_k^T + R_k\,)^{-1} \tag{4.2}$$

$$K^\theta_{k|k} = \tilde{K}^\theta_k \tag{4.3}$$

$$\begin{aligned} \hat{\theta}_{k+1|k} &= \left[[\hat{\theta}^1_A(k+1|k)]^T \mid \ldots \mid [\hat{\theta}^p_A(k+1|k)]^T \mid [\hat{\theta}^1_B(k+1|k)]^T \mid \ldots \mid [\hat{\theta}^m_B(k+1|k)]^T\right]^T \\ &= \hat{\theta}_{k|k-1} + K^\theta_{k|k}\,(\,z_k - H_k\,\hat{s}_{k|k-1}\,) \end{aligned} \tag{4.4}$$

$$E_{\psi_k}(\,G^x_k\,) = E\,(\,G^x_k \mid \psi_k\,) = G^x(\,\hat{\theta}_{k+1|k}\,) \tag{4.5}$$

$$K^x_{k|k} = \tilde{K}^x_k + E_{\psi_k}(G^x_k)\,S_k\;(H_k\,P_{k|k-1}\,H_k^T + R_k\,)^{-1} \tag{4.6}$$

$$\hat{x}_{k+1|k} = F^x_k\,\hat{s}_{k|k-1} + K^x_{k|k}\,(\,z_k - H_k\,\hat{s}_{k|k-1}\,) \tag{4.7}$$

$$\begin{aligned} \tilde{P}_{k+1} = \begin{bmatrix} \tilde{P}^{xx}_{k+1} & \tilde{P}^{x\theta}_{k+1} \\ \tilde{P}^{\theta x}_{k+1} & \tilde{P}^{\theta\theta}_{k+1} \end{bmatrix} &= F_k\,P_{k|k-1}\,F_k^T + E_{\psi_k}(G)\,Q_0(k)\,E_{\psi_k}(G^T) \\ &\quad - K_{k|k}\,(\,H_k\,P_{k|k-1}\,H_k^T + R_k\,)K^T_{k|k} \end{aligned} \tag{4.8}$$

$$\left[P^{\theta x}_{k+1|k}\right]^T = P^{x\theta}_{k+1|k} = \tilde{P}^{x\theta}_{k+1} \tag{4.9}$$

$$P^{\theta\theta}_{k+1|k} = \tilde{P}^{\theta\theta}_{k+1} \tag{4.10}$$

$$\begin{aligned} P^{xx}_{k+1|k} = \tilde{P}^{xx}_{k+1} &+ \sum_{i=1}^{p}\sum_{j=1}^{p} r_{i,j}(k)\;P^{\theta\theta}_{i,j}(k+1|k) \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{m} q_{i,j}(k)\;P^{\theta\theta}_{p+i,p+j}(k+1|k) \end{aligned} \tag{4.11}$$

In Equation 4.11, $P^{\theta\theta}_{k+1|k}$ has been partitioned into $n \times n$ submatrices according to

$$P^{\theta\theta}_{k+1|k} = \begin{bmatrix} P^{\theta\theta}_{1,1}(k+1|k) & & P^{\theta\theta}_{1,p+m}(k+1|k) \\ \vdots & \cdots & \vdots \\ P^{\theta\theta}_{p+m,1}(k+1|k) & & P^{\theta\theta}_{p+m,p+m}(k+1|k) \end{bmatrix} \tag{4.12}$$

where $P^{\theta\theta}_{i,j}(k+1|k) = E\{\,[\,\hat{\theta}^i_A(k+1|k) - \theta^i_A\,][\,\hat{\theta}^j_A(k+1|k) - \theta^j_A\,]^T \mid \psi_k\}$,
$P^{\theta\theta}_{p+i,\,p+j}(k+1|k) = E\{\,[\,\hat{\theta}^i_B(k+1|k) - \theta^i_B\,][\,\hat{\theta}^j_B(k+1|k) - \theta^j_B\,]^T \mid \psi_k\}$, *etc.*, (see § 5).

5. Optimality of the PLID Algorithm

Theorem 5.1: For MIMO stochastic time-invariant systems of the class described in Section 1, under standard gaussian assumptions on the noise vectors and initial state, PLID is the optimal estimator (in the mean square sense) of the states and parameters, given only the measurement data.

Proof: Recall that $\{\,\psi_k\,\}$ is the increasing sequence of σ-fields generated by the increasing set of measurement data, $\{\,z_0\,, \ldots, z_k\,\}$.

Consider the stochastic system $\overline{S}$ of Equation 1.7. Suppose the input noise vector v_k and output noise vector w_k are independent gaussian white noise vectors with zero means and known covariances, and the parameter noise vectors $\Xi^i_A(k)$ and $\Xi^j_B(k)$ are identically zero for all i,j,k (*i.e.*, the parameters are constant). Further suppose that the initial state s_0 is gaussian distributed. Clearly, since the underlying system (given by Equations 1.2 and 1.3) is linear time-invariant, and the input and output noises are gaussian, the vectors x_k and z_k are jointly gaussian.

Since the parameter vector θ_k is a constant, it follows that s_k and z_k are also jointly gaussian (easily shown by considering the characteristic function). Therefore, the *linear* conditional minimum variance estimate of s_k relative to ψ_k is also *the* conditional minimum variance estimator of s_k relative to ψ_k [4], which is the optimal estimator, in the mean-square error sense, relative to ψ_k.

The PLID algorithm is derived in Section 2 as the linear conditional minimum variance estimator of the extended state vector s_k, relative to ψ_k. Therefore, under the gaussian assumptions, it is *the* conditional minimum variance estimator relative to ψ_k. However, the full specification of the algorithm involves a linear function of the *conditional mean* estimate of the parameters (see Equation 4.6) and a linear function of the *conditional parameter error covariance matrix* (see Equation 4.11).

But the conditional minimum mean square estimate relative to ψ_k is *equivalent* to the conditional mean estimate relative to ψ_k [4], so Equations 4.6 and 4.11 can be computed using the PLID parameter estimates $\hat{s}_{k+1|k}$ and the PLID parameter error covariance $P^{\theta\theta}_{k+1|k}$. Therefore, under standard gaussian assumptions, PLID is the optimal estimator of the states and parameters, in the mean square sense, given only the measurement data. ■

6. Comparison of PLID and the Extended Kalman Filter

Figures 1 and 2 present a comparison of the parameter estimates of PLID and the Extended Kalman Filter (EKF), when the initial estimate is not close to the actual parameter values. The SISO system being simulated has a near pole-zero cancellation, and a non-minimum-phase zero. The figures were obtained with a signal-to-noise ratio of about 30 dB at the input and at the output. The state estimates are not shown, but those of the PLID algorithm are correspondingly better than those of the EKF. These results should not be surprising in view of the fact that PLID is the optimal estimator of the states and parameters.

7. Summary

The pseudo-linear identification (PLID) algorithm was developed from first principles. PLID can be used for joint parameter and state estimation of linear, completely controllable and observable, stochastic MIMO systems with known structure The time-optimality of PLID was proved for the deterministic case; the *optimality* of PLID was proved for the stochastic case.

The stochastic PLID algorithm is an extension of earlier work by Salut, *et.al.* [1], to the more useful case where there is input noise and measurement noise. (If Salut's method is applied to such a case, biased estimates result.)

Convergence a.e. of the stochastic PLID parameter estimates to the true parameter values has also been proved (under the same gaussian assumptions as those above), but that proof will be published elsewhere.

References

[1] Salut, Gerard, J. Aguilar-Martin, and S. Lefebvre, "New Results on Optimal Joint Parameter and State Estimation of Linear Stochastic Systems," *Transactions of the ASME*, Vol. 102, pp. 28-34, March 1980.

[2] VanLandingham, Hugh, and Mark Hopkins, "Deadbeat Parameter Identification of DARMA Processes," *Proc. 24th Annual Conf. on Communication, Control, and Computing (Allerton)*, University of Illinois at Urbana-Champaign, 1986, pp. 641-642.

[3] Chen, Chi-Tsong, *Introduction to Linear System Theory*, New York: Holt, Rinehart and Winston, Inc., 1970, pp.268-270.

[4] Anderson, B.D.O., and J.B. Moore, *Optimal Filtering*, Englewood Cliffs NJ: Prentice-Hall, Inc., 1979, p. 94, and pp. 26-27.

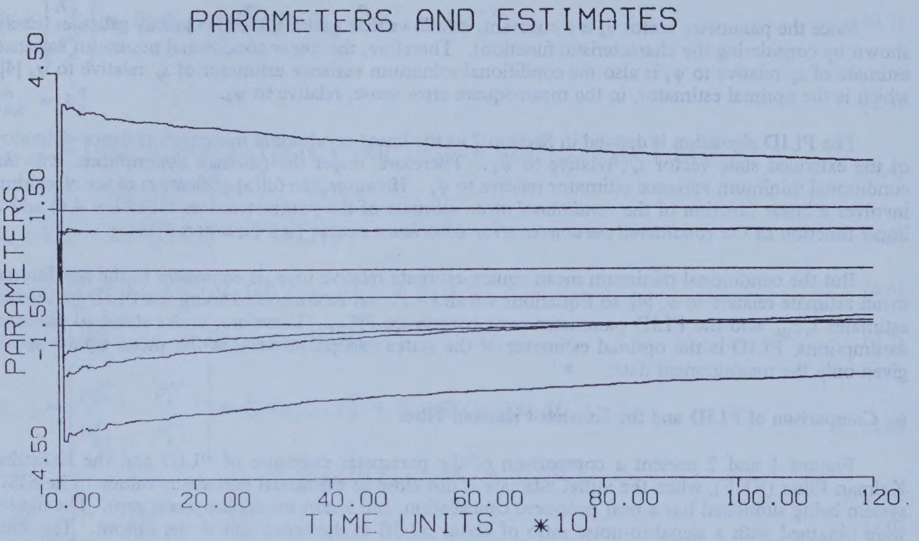


Figure 1. EKF converges beyond 1200 iterations. Maximum initial estimate error is about 3.0.

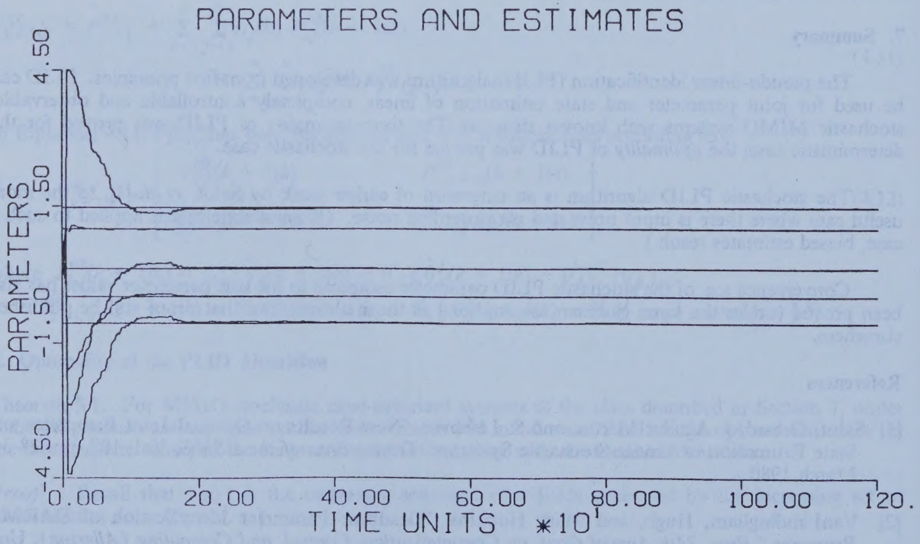


Figure 2. PLID converges in about 200 iterations. Maximum initial estimate error is about 3.5.

A HYBRID METHOD OF ESTIMATING THE PARAMETERS OF ARMA MODELS

SHIPING LI and YAO ZHU
Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843

BRADLEY W. DICKINSON
Department of Electrical Engineering
Princeton University
Princeton, New Jersey 08544

ABSTRACT

A hybrid method for estimating the parameters of ARMA models is presented. Based on analysis and on simulation experiments involving two methods developed recently, we propose to combine their advantages to form this hybrid estimation method. With this new method, we can get the estimates of the AR and MA parameters with greater accuracy by just solving linear equations.

I. INTRODUCTION

It is well known that LMS autoregressive (AR) modeling is a linear problem, and that LMS autoregressive-moving average (ARMA) modeling is a nonlinear one. It is also known that a finite order stationary and minimum phase ARMA(p,q) model is equivalent to an infinite order AR model, that is, the process $x(n)$ described by

$$\left[1+\sum_{i=1}^{p} a_i z^{-i}\right] x(n) = \left[1+\sum_{j=1}^{q} b_j z^{-j}\right] w(n) \tag{1}$$

is equivalent to that described by

$$\left[1+\sum_{k=1}^{\infty} a_k^{\infty} z^{-k}\right] x(n) = w(n) \tag{2}$$

where z^{-1} is the delay operator, $z^{-1}x(n) = x(n-1)$, $w(n)$ is a white noise process with zero mean and variance σ_w^2. If we define

$$A(z) = 1+\sum_{i=1}^{p} a_i z^{-i} \qquad B(z) = 1+\sum_{j=1}^{q} b_j z^{-j} \qquad A^{\infty}(z) = 1+\sum_{k=1}^{\infty} a_k^{\infty} z^{-k}$$

then $A(z)$, $B(z)$ and $A^{\infty}(z)$ are related through

$$A^{\infty}(z) = B^{-1}(z)A(z) \tag{3}$$

Two methods of estimating the parameters of the ARMA(p,q) model are based on or closely related to this equivalence relation. One method was derived directly from the equivalence relation by Graupe, Krause and Moore [1]. The other method was derived by Li and Dickinson [2,3,4] based on an iterated LMS regression approach. The end results bear close resemblance to that of Graupe *et al.* It is interesting to compare these two methods, and to combine the advantages of both methods in forming an improved method for the ARMA modeling problem.

II. A COMPARISON OF TWO METHODS OF ARMA MODELING

Based on the above equivalence relation, Graupe *et al* [1] showed that the MA parameters b_j, $1 \le j \le q$, can be obtained from the following *linear* equations

$$\begin{bmatrix} a_m^{\infty} & a_{m-1}^{\infty} & \cdots & a_{m-q+1}^{\infty} \\ a_{m+1}^{\infty} & a_m^{\infty} & \cdots & a_{m-q}^{\infty} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m+q-1}^{\infty} & a_{m+q-2}^{\infty} & \cdots & a_m^{\infty} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_q \end{bmatrix} = -\begin{bmatrix} a_{m+1}^{\infty} \\ a_{m+2}^{\infty} \\ \vdots \\ a_{m+q}^{\infty} \end{bmatrix} \tag{4}$$

If $m \geq p$, then $c_j = b_j$, $1 \leq j \leq q$. The AR parameters a_i, $1 \leq i \leq p$, are given by

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} a_1^\infty \\ a_2^\infty \\ \vdots \\ a_p^\infty \end{bmatrix} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^\infty & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{p-1}^\infty & a_{p-2}^\infty & \cdots & a_{p-q}^\infty \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} \tag{5}$$

This result is interesting in that it transforms a nonlinear problem into a linear one. However, this new linear problem is not realizable without some approximations. It is not possible to obtain the parameters of an infinite order AR model, and some approximations have to be made by using a high order AR model instead of an infinite order model. The question is how good the approximation is if all the a_i^∞'s in (4) and (5) are replaced by the a_i^h's of an hth order AR(h) model with $h > (p+q)$, where the AR(h) model is the best hth order approximation to the AR(∞) in the LMS sense.

Recently, it was shown by Li and Dickinson [2,3,4] that the parameters of an ARMA(p,q) model can also be obtained by first obtaining a $(p+q)$-order AR model and then solving a set of linear equations similar to those of Graupe *et al.* Specifically, this method consists of first solving q linear equations for a set of auxiliary variables β_j's,

$$\begin{bmatrix} a_p^{p+q-1} & a_{p-1}^{p+q-2} & \cdots & a_{p-q+1}^p \\ a_{p+1}^{p+q-1} & a_p^{p+q-2} & \cdots & a_{p-q+2}^p \\ \vdots & \vdots & \ddots & \vdots \\ a_{p+q-1}^{p+q-1} & a_{p+q-2}^{p+q-2} & \cdots & a_q^p \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_q \end{bmatrix} = - \begin{bmatrix} a_{p+1}^{p+q} \\ a_{p+2}^{p+q} \\ \vdots \\ a_{p+q}^{p+q} \end{bmatrix} \tag{6}$$

Then the AR parameters a_i, $1 \leq i \leq p$, are given by

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_q \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} a_1^{p+q} \\ a_2^{p+q} \\ \vdots \\ a_q^{p+q} \\ \vdots \\ a_p^{p+q} \end{bmatrix} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^{p+q-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{q-1}^{p+q-1} & a_{q-2}^{p+q-2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{p-1}^{p+q-1} & a_{p-2}^{p+q-2} & \cdots & a_{p-q}^p \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_q \end{bmatrix} \tag{7}$$

and the MA parameter b_j's are approximated by

$$\hat{b}_j = \left[\prod_{i=1}^{j} (1 - K_{p+q-i}^2) \right]^{-\frac{1}{2}} \beta_j, \qquad 1 \leq j \leq q \tag{8}$$

Here K_m, $1 \leq m \leq p+q$, are the first $(p+q)$ partial correlation (PARCOR) coefficients of the ARMA(p,q) or AR(∞) model, and a_k^m is the kth coefficients of the AR(m) model that is the best mth order approximation of the AR(∞) model. Note that parameters from different orders are used. All these a_k^m's can be obtained from the $p+q$ PARCOR parameters K_m, $1 \leq i \leq p+q$, using the Levinson recursion,

$$a_k^{m+1} = \begin{cases} 1 & i = 0 \\ a_k^m + K_{m+1} a_{m-k+1}^m & 1 \leq k \leq m \\ K_{m+1} & k = m+1 \end{cases} \tag{9}$$

An estimation method based on (6)-(9) is readily implementable because knowledge of the infinite order AR model is not required. All that is needed is to first estimate a $(p+q)$ order lattice filter, that is, the PARCOR parameters K_i, $1 \leq i \leq p+q$, then obtain the a_k^m's from these PARCOR parameters using (9), and finally the AR and MA parameter estimates are obtained by using (6), (7) and (8). It was shown that the AR estimates so obtained are consistent and the simulation results also indicate excellent performance. The MA estimates are only approximations of the true parameters. They are somewhat biased, especially for the pure MA or strongly MA dominated ARMA models.

Although the a_i, b_j and a_k^∞ parameters are related linearly, the high order approximation a_k^h's of a_k^∞'s does not preserve the linear relation with the a_i's and b_j's. Actually the general form of the exact relation is too complicated to put into compact equations. Therefore, it is very difficult to obtain further analytic comparison of these two methods. However, we can easily compare them by simulation experiments. Here are two examples.

Example 1: Consider the ARMA(4,2) model given by

$$\left[1 + 0.479z^{-1} + 0.086z^{-2} + 0.29z^{-3} + 0.731z^{-4}\right]x(n) = \left[1 + 0.723z^{-2}\right]w(n)$$

which corresponds to two pairs of poles located at $0.95e^{\pm j0.8\pi}$, $0.9e^{\pm j0.3\pi}$ and one pair of zeros located at $0.85e^{\pm j0.5\pi}$. A long sequence of data was generated with this model, then we selected 10 consecutive segments each with 600 data points from the sequence, and used each segment in simulations of the two methods respectively. The estimates from each simulation were converted to corresponding poles and zeros, and all ten sets of results are plotted in Figure 1 and Figure 2 respectively. From these two figures it is clear that for the AR parameters, Li and Dickinson's method gives better estimates than Graupe *et al*'s method; and for the MA parameters, estimates given by Graupe *et al*'s method are better. Here for Graupe *et al*'s method, we chose $h = 25$, which seems to be the best choice in the approximation $a_i^h \approx a_i^\infty$ for this example. The contrast is even greater in the next example with a different pole-zero configuration.

Example 2. In this example we use the following model

$$\left[1 - 2.595z^{-1} + 3.339z^{-2} - 2.2z^{-3} + 0.731z^{-4}\right]x(n) = \left[1 + 0.556z^{-1} + 0.81z^{-2}\right]w(n)$$

which corresponds to two pairs of poles located at $0.95e^{\pm j0.2\pi}$, $0.9e^{\pm j0.3\pi}$ and one pair of zeros located at $0.9e^{\pm j0.6\pi}$. Ten sets of simulation results using the two methods are plotted in Figure 4 and Figure 5 respectively. These figures clearly indicate that Li and Dickinson's method gives superior AR estimates, however, the MA estimates are conservative in the way that they are biased towards the origin, away from the true zero positions; On the other hand, although the AR estimates from Graupe *et al*'s method are fairly poor, the MA estimates are good, and they have no apparent bias.

These two examples and our extensive simulation experiments indicate that if poles and zero are located far apart from each other, both methods give good estimates; however, if poles and zeros are clustered closely together, these two methods perform differently. For the AR parameters, Li and Dickinson's method gives superior estimates to those given by Graupe *et al*'s method; and for MA parameters, it's the other way around. Our simulations also reveal one problem with the choice of the order h in using Graupe's method. Although theoretically, the higher the order h, the closer the approximations, choice of large h also has some undesirable effects. Because of the statistical nature of the estimates, eventually larger h means noisier estimates $\hat{a}_i^h$'s for a_i^h's, which transform to poorer estimates of the true ARMA parameters, in addition to resulting in increased complexity.

III. WAYS TO IMPROVE THESE TWO METHODS

One way to reduce the noisy character of $\hat{a}_i^h$'s in using Graupe's method is by using some type of average. Consider the estimated high order approximation of equation (4),

$$\begin{bmatrix} \hat{a}_m^h & \hat{a}_{m-1}^h & \cdots & \hat{a}_{m-q+1}^h \\ \hat{a}_{m+1}^h & \hat{a}_m^h & \cdots & \hat{a}_{m-q}^h \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{m+q-1}^h & \hat{a}_{m+q-2}^h & \cdots & \hat{a}_m^h \end{bmatrix} \begin{bmatrix} c_1^h \\ c_2^h \\ \vdots \\ c_q^h \end{bmatrix} = - \begin{bmatrix} \hat{a}_{m+1}^h \\ \hat{a}_{m+2}^h \\ \vdots \\ \hat{a}_{m+q}^h \end{bmatrix} \tag{10}$$

Since the order h can be chosen freely, one approach is to choose several values for h, say, $h_1, h_2, \cdots, h_M$, then take an average over these M equations to obtain

$$\begin{bmatrix} \hat{a}_m & \hat{a}_{m-1} & \cdots & \hat{a}_{m-q+1} \\ \hat{a}_{m+1} & \hat{a}_m & \cdots & \hat{a}_{m-q} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{m+q-1} & \hat{a}_{m+q-2} & \cdots & \hat{a}_m \end{bmatrix} \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_q \end{bmatrix} = - \begin{bmatrix} \hat{a}_{m+1} \\ \hat{a}_{m+2} \\ \vdots \\ \hat{a}_{m+q} \end{bmatrix} \tag{11}$$

where

$$\hat{a}_k = \frac{1}{M}\left[\hat{a}_k^{h_1} + \hat{a}_k^{h_2} + \cdots + \hat{a}_k^{h_M}\right] \qquad m - q + 1 \leq k \leq m + q + 1$$

Another approach is to use a fixed h, while choosing a number of different values for m, say, $m_1, m_2, \cdots, m_L$, all greater than p, then average on these L equations to obtain

$$\begin{bmatrix} \hat{a}_0^h & \hat{a}_{-1}^h & \cdots & \hat{a}_{1-q}^h \\ \hat{a}_1^h & \hat{a}_0^h & \cdots & \hat{a}_{-q}^h \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{q-1}^h & \hat{a}_{q-2}^h & \cdots & \hat{a}_0^h \end{bmatrix} \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \\ \vdots \\ \bar{c}_q \end{bmatrix} = - \begin{bmatrix} \hat{a}_1^h \\ \hat{a}_2^h \\ \vdots \\ \hat{a}_q^h \end{bmatrix} \tag{12}$$

with

$$\hat{a}_i^h = \frac{1}{L}\left[\hat{a}_{m_1+i}^h + \hat{a}_{m_2+i}^h + \cdots + \hat{a}_{m_L+i}^h\right] \qquad -q + 1 \leq i \leq q$$

Although these $\hat{a}_k^h$'s are not statistically uncorrelated, one would expect that the averaging operation would give some improvement in the estimates. Our computer simulations showed limited improvement in the estimates using the above suggested averaging techniques with the Graupe *et al* method.

Another approach to improve the ARMA parameter estimates is to combine the two methods together to form a hybrid method. Specifically, by combining the AR estimates from Li and Dickinson's method and the MA estimates from Graupe *et al*'s method into this hybrid ARMA estimator, we can get both accurate AR estimates and accurate MA estimates *linearly*. The computational complexity of this hybrid method is that of Graupe *et al*'s method plus that of solving equations (6) and (7). This is less than the sum of the total computation of both methods because both are based on the first h PARCOR parameters and the Levinson recursion.

In Figure 3 and Figure 6 we have plotted the poles and zero estimates using the above suggested hybrid ARMA estimation method for the two examples given earlier. These demonstrate the improvements that are achieved by the hybrid method.

In conclusion, the hybrid method, although just a simple combination of two known methods, gives superior estimates of both the AR and MA parameters of an ARMA model. The advantages of this method, like its parent methods, is that it is still a linear method, and can be obtained by first estimating the first h (about two or three times $p + q$ is a good choice) PARCOR parameters K_i's, then converting these K_i's to the corresponding parameters a_{ik}^k, and finally solving (6) and (7) for the AR parameter estimates and solving (10) for the MA parameter estimates.

REFERENCES

[1] D. Graupe, D. J. Krause, and J. B. Moore, "Identification of autoregressive moving-average parameters of time series," *IEEE Transactions on Automatic Control,* vol. AC-20, pp. 104-107, Feb. 1975.

[2] S. Li and B. W. Dickinson, "An efficient method to compute consistent estimates of the AR parameters of an ARMA model," *IEEE Transactions on Automatic Control,* vol. AC-31, pp. 275-278, March 1986.

[3] S. Li, "Applications of lattice filters in estimation of time series models," Ph.D Dissertation, Princeton University, Princeton, New Jersey, Oct. 1986.

[4] S. Li and B. W. Dickinson, "Application of the lattice filter to robust estimation of AR and ARMA models," to appear in *IEEE Transactions on Acoustics, Speech and Signal Processing.*

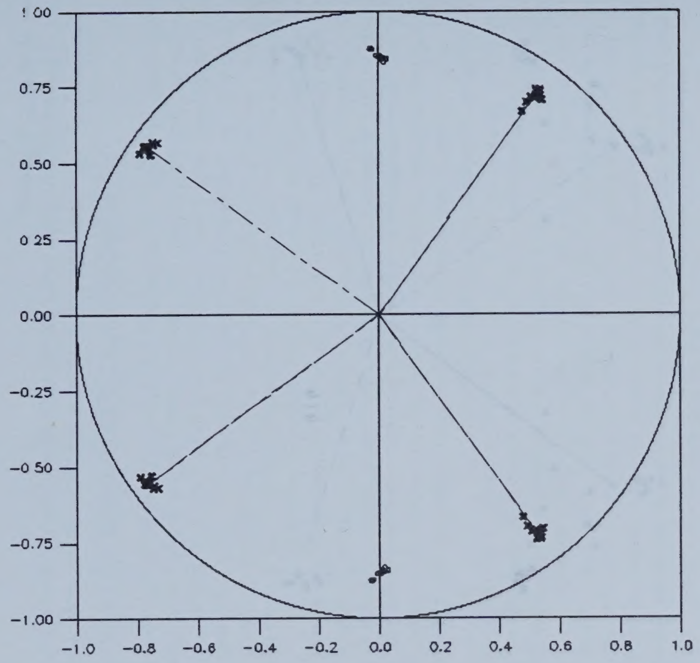


Figure 1. Pole-zero estimates using Graupe *et al*'s method from ten simulation results, each with 600 data from an ARMA(4,2) process with poles located at $0.95e^{\pm j0.8\pi}, 0.9e^{\pm j0.3\pi}$ and zero at $0.85e^{\pm j0.5\pi}$.

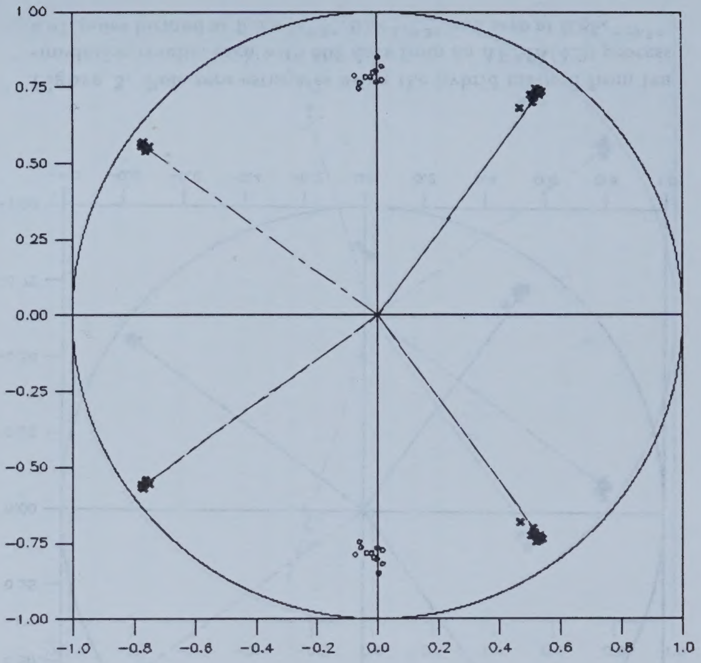


Figure 2. Pole-zero estimates using Li and Dickinson method from ten simulation results, each with 600 data from an ARMA(4,2) process with poles located at $0.95e^{\pm j0.8\pi}, 0.9e^{\pm j0.3\pi}$ and zero at $0.85e^{\pm j0.5\pi}$.

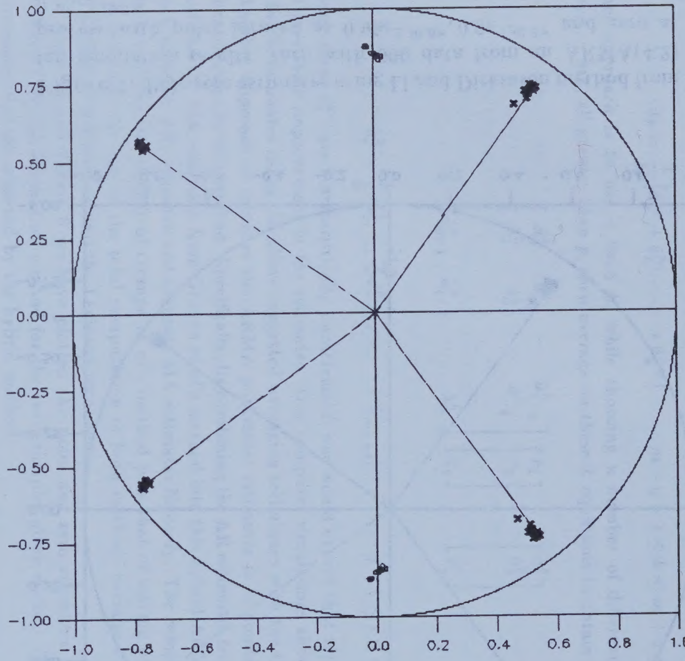


Figure 3. Pole-zero estimates using the hybrid method from ten simulation results, each with 600 data from an ARMA(4,2) process with poles located at $0.95e^{\pm j0.8\pi}$, $0.9e^{\pm j0.3\pi}$ and zero at $0.85e^{\pm j0.5\pi}$.

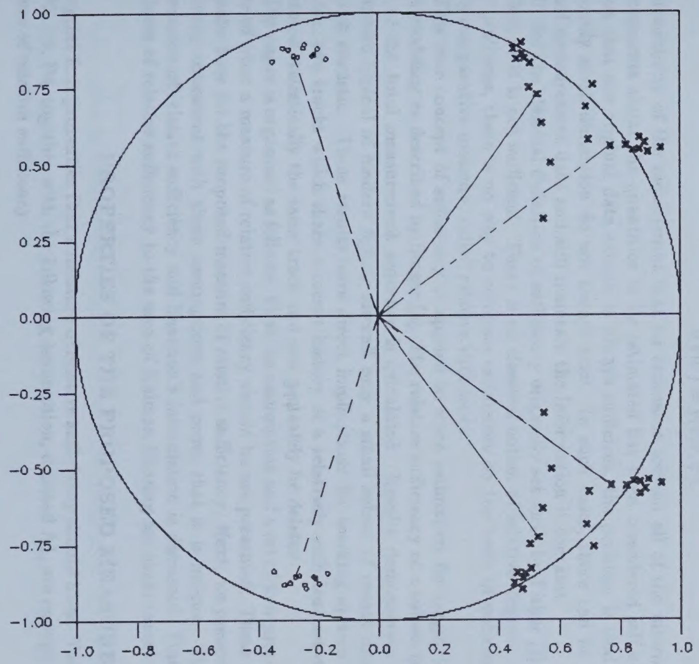


Figure 4. Pole-zero estimates using Graupe *et al*'s method from ten simulation results, each with 600 data from an ARMA(4,2) process with poles located at $0.95e^{\pm j0.2\pi}, 0.9e^{\pm j0.3\pi}$ and zero at $0.9e^{\pm j0.6\pi}$.

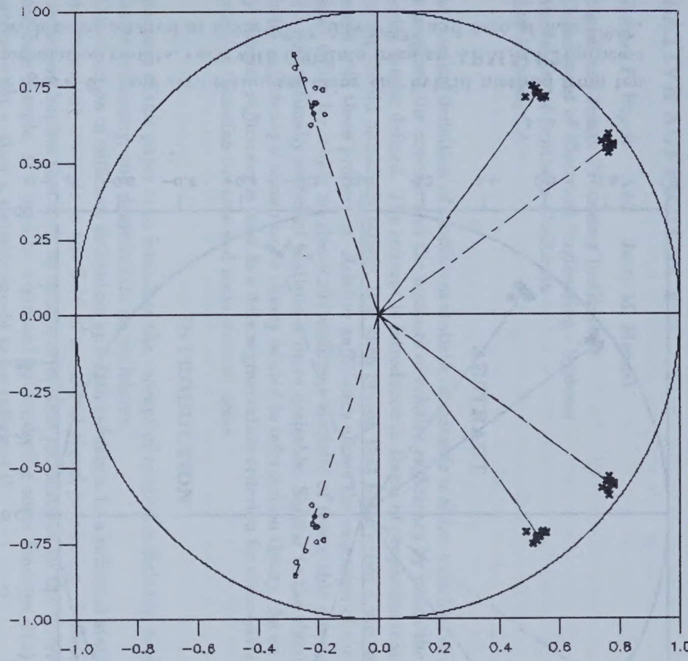


Figure 5. Pole-zero estimates using Li and Dickinson method from ten simulation results, each with 600 data from an ARMA(4.2) process with poles located at $0.95e^{\pm j0.2\pi}, 0.9e^{\pm j0.3\pi}$ and zero at $0.9e^{\pm j0.6\pi}$.

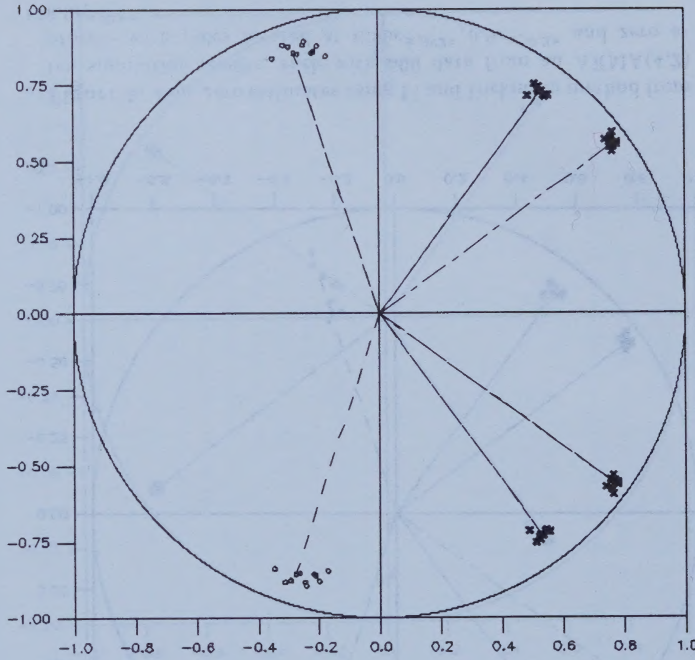


Figure 6. Pole zero estimates using the hybrid method from ten simulation results, each with 600 data from an ARMA(4,2) process with poles located at $0.95e^{\pm j0.2\pi}, 0.9e^{\pm j0.3\pi}$ and zero at $0.9e^{\pm j0.6\pi}$.

RELATIVE SUFFICIENCY

Robert F. Popoli And Jerry M. Mendel

Signal and Image Processing Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA

ABSTRACT

By the classical definition of a sufficient statistic, a statistic is either sufficient or it is not sufficient. This paper proposes the concept and the measure of *relative sufficiency* by which the degree of sufficiency of a statistic can be defined. The concept is introduced in terms of properties that the measure of *relative sufficiency* should intuitively satisfy. It is then proven that there exists a unique measure of sufficiency consistent with these properties. *Relative sufficiency* allows the extension of the notion of sufficiency to problems either for which no classically sufficient statistic of reasonable dimension exists or problems where a sub-optimally sufficient statistic is more desirable. Similar to the classical notion, the concept of *relative sufficiency* is shown to be closely related to information theory. As an example of its practical utility, *relative sufficiency* is used as a data segmentation criterion for state-variable system models with independent Gaussian process and measurement noise.

INTRODUCTION

The purpose of this paper is to introduce the concept of relative sufficiency, establish a unique measure of relative sufficiency, and demonstrate its usefulness.

Construction of sufficient estimators and data reduction to a sufficient statistic for further processing are important concepts in the field of statistical signal processing. Sufficiency of an estimator or data reduction implies that such an estimator or data set contains all the "relevant information" of the original data sample. If θ is to be estimated by employing some function $\psi(z)$ (i.e. a statistic) of the measurement set z, then a statistic is said to be sufficient if

$$p(\theta|\psi) = p(\theta|z,\psi). \tag{1}$$

The desirability of the requirement that an estimator contain all of the information provided in the measurements about the quantities to be estimated has been considered self evident [1]. While it is obvious that the original data sample is always sufficient, the problem is that sufficient statistics of reasonably small dimension do not always exist. In such cases, there can be no way to reduce the original measurement data and still maintain the information it contains.

Under the classical definition of sufficiency originally set forth by Fisher [2], an estimator is either sufficient or it is not sufficient. There is no classical notion of relative degrees of sufficiency. Thus for many problems, there is no way to compare estimators on the basis of sufficiency. This paper offers such a comparative measure, called relative sufficiency.

This new concept of sufficiency is applied to state estimation for target tracking employing the target modeling as described by Singer [3]. The relative sufficiency of a subset (representing the recent past) of the total measurement sequence is calculated. Results demonstrate that for target model parameters typical of modern fighter aircraft only a small subset of recent measurements is a nearly sufficient statistic. These results have direct implications for tracking system implementations. For example, two tracks which share a recent history of a relatively sufficient set of observations may be considered statistically the same track and one can safely be deleted.

The paper is organized as follows. First, an assumption and a set of four propositions concerning the nature of what a measure of relative sufficiency should be are presented. These properties serve as the axiomatic base for the proposed measure of relative sufficiency. Next, we present a measure of relative sufficiency consistent with these assumptions and prove that it is unique. The connection between our measure of relative sufficiency and Shannon's information is discussed. Finally, an example of the usefulness of relative sufficiency to the area of Kalman filtering for multi-target tracking is discussed.

PROPERTIES OF THE PROPOSED MEASURE

We suggest four properties that a measure of relative sufficiency should have. These properties, denoted P1, P2, P3, P4, together with the following assumption, denoted A1, are enough to determine a unique measure of relative sufficiency.

A1′) We assume that the measure of relative sufficiency, denoted $S(\psi;\Theta,Z)$, is expressible as the average over the joint probability space of θ and z (denoted $\Theta \times Z$) of some function, say $f(\circ)$, of the posterior density $p(\theta|\psi)$.

A1′ has the following motivation. As seen from Equation (1), the classical definition of sufficiency is a condition imposed on the posterior density $p(\theta|\psi)$. It is desirable that any measure of a relative sufficient statistic for a given problem should also be a functional of this density. The idea behind assumption A1′ is that the relative sufficiency of a statistic $\psi(z)$ should be an indication of its average behavior over the ensemble of realizations of the random parameter θ and measurements z.

Actually, we make the following stronger statement of assumption A1′.

A1) We assume that the measure of relative sufficiency with respect to any given event E in the $\Theta \times Z$ space, denoted $S_E(\psi;\Theta,Z)$, is expressible as the average of $f(p(\theta|\psi)$ over that event.

This stronger statement of A1′ expresses the idea that the measure of relative sufficiency should exist even if we wish to restrict our attention to *any* given event in the $\Theta \times Z$ space.

Given assumption A1, we are now in a position of seeking the function $f(\circ)$. We suggest that this function should be selected to assure that the following four properties are satisfied.

P1) The function f should be continuous in its argument.

P2) The *relative sufficiency* should lie in the interval [0,1].

P3) The *relative sufficiency* should achieve its maximum, unity, when ψ is sufficient in the classical sense. The *relative sufficiency* should achieve its minimum, zero, when ψ is independent of θ.

P4) Iff two statistics ψ_1 and ψ_2 are independent (i.e. they are both unconditionally and conditionally independent with respect to θ) then the *relative sufficiency* should satisfy

$$S(\psi_1,\psi_2;\Theta,Z) = S(\psi_1;\Theta,Z) + S(\psi_2;\Theta,Z)$$

P1 is a way to impose a regularity constraint on the measure of relative sufficiency. The continuity of f, required by P1, is desirable since we would like small changes in the value of $p(\theta|\psi)$ for a given θ and z (hence $\psi(z)$) to result in small changes in the relative sufficiency.

P2 is simply a convenient selection of range for the value of relative sufficiency.

P3 gives obvious conditions under which the measure of relative sufficiency should achieve its extreme values. We would like a relative sufficiency of unity to indicate that a statistic ψ is sufficient in the classical sense, and a relative sufficiency of zero to indicate that ψ is statistically unlinked to θ.

Finally, P4 gives the conditions under which we would expect the relative sufficiency of a joint statistic (ψ_1,ψ_2) to be equal to the sum of the relative sufficiencies of its parts. For example, if we partition a measurement set consisting of N independent measurements into two sets called statistics ψ_1 and ψ_2, we would expect that the joint statistic (i.e. the original data set) should have sufficiency $1 = S(\psi_1;\Theta,Z) + S(\psi_2;\Theta,Z)$.

DERIVATION OF THE MEASURE OF RELATIVE SUFFICIENCY

The central goal of this paper is to establish the fact that assumption A1 and properties P1 through P4 are sufficient to uniquely determine the exact functional form of $S(\psi;\Theta,Z)$ as

$$S(\psi;\Theta,Z) = \frac{\iint p(\theta,z)\log\left[\frac{p(\theta|\psi)}{p(\theta)}\right] d\theta dz}{\iint p(\theta,z)\log\left[\frac{p(\theta|z)}{p(\theta)}\right] d\theta dz} \tag{2}$$

PROOF: Under P4, we require that if ψ_1 and ψ_2 are independent statistics,

$$S(\psi_1,\psi_2;\Theta,Z) = S(\psi_1;\Theta,Z) + S(\psi_2;\Theta,Z) \tag{3}$$

By A1 and P1, Eq. (3) can be written explicitly as a condition on the posterior densities, as

$$\iint_E f(p(\theta|\psi_1,\psi_2))p(\theta,z)d\theta dz = \iint_E f(p(\theta|\psi_1))p(\theta,z)d\theta dz + \iint_E f(p(\theta|\psi_2))p(\theta,z)d\theta dz \tag{4}$$

But under the strong assumption A1, Eq. (4) must be true for all events (i.e., measurable subsets of $\Theta \times Z$). This implies that we require

$$f(p(\theta|\psi_1, \psi_2)) = f(p(\theta|\psi_1)) + f(p(\theta|\psi_2)) \tag{5}$$

We next consider the implication of Eq. (5) on the functional form of $f(\circ)$. Define a function $g(\circ) = f(\circ p(\theta))$ or equivalently $f(\circ) = g\left(\frac{\circ}{p(\theta)}\right)$. Then, from Eq. (5) we want

$$g\left(\frac{p(\theta|\psi_1, \psi_2)}{p(\theta)}\right) = g\left(\frac{p(\theta|\psi_1)}{p(\theta)}\right) + g\left(\frac{p(\theta|\psi_2)}{p(\theta)}\right) \tag{6}$$

From Bayes' rule, Eq. (6) becomes

$$g\left(\frac{p(\psi_1, \psi_2|\theta)}{p(\psi_1, \psi_2)}\right) = g\left(\frac{p(\psi_1|\theta)}{p(\psi_1)}\right) + g\left(\frac{p(\psi_2|\theta)}{p(\psi_2)}\right) \tag{7}$$

Under the independence assumption of P4, the left hand side of Eq. (7) can be rewritten as

$$g\left(\frac{p(\psi_1|\theta)}{p(\psi_1)}\frac{p(\psi_2|\theta)}{p(\psi_2)}\right) = g\left(\frac{p(\psi_1|\theta)}{p(\psi_1)}\right) + g\left(\frac{p(\psi_2|\theta)}{p(\psi_2)}\right) \tag{8}$$

The only continuous non-zero function which satisfies Eq. (8) is the logarithm [4]; thus, we have deduced that $g(\circ) = \log[\circ]$ (where the base is unspecified) or that

$$f(\circ) = \log\left[\frac{\circ}{p(\theta)}\right] \tag{9}$$

So far, we have shown that the measure of relative sufficiency with respect to any given event E must be given to within a scale constant K by

$$S_E(\psi; \Theta, Z) = K \iint_E \log\left[\frac{p(\theta|\psi)}{p(\theta)}\right] p(\theta, z) d\theta dz \tag{10}$$

All that remains is find the constant K such that P2 and P3 are satisfied. It is easy to verify that K is given by

$$K = \frac{1}{\iint_E \log\left[\frac{p(\theta|z)}{p(\theta)}\right] p(\theta, z) d\theta dz} \tag{11}$$

Note that, although problem dependent information about θ and z help to determine K, in no way does the statistic ψ enter into the calculation of K.

THE CONNECTION TO INFORMATION THEORY

As with the classical concept of sufficiency, the ties between relative sufficiency and information theory are strong. In fact, Eq. (2) is the ratio of the mutual information between the statistic ψ and θ to the mutual information between the data set z and θ. Thus

$$S(\psi; \Theta, Z) = \frac{I(\Theta; \psi(Z))}{I(\Theta; Z)} \tag{12}$$

From this information theoretic vantage, the intuitive appeal of this ratio is obvious. Kullback [5] recognized that such a ratio represented the percentage of information retained by a statistic, and hence, indicated its effectiveness to discriminate between two hypotheses. Our contribution is that we have shown that the importance of Eq. (2) need not be accepted on the basis of information theoretic intuition; but rather, Eq. (2) is the unique result of an axiomatic formulation of the concept of relative sufficiency. In a sense, we have built our measure of relative sufficiency "from the ground up". Our axiomatic development of Eq.(2) establishes its appropriateness and uniqueness as the measure of relative sufficiency.

MORE ON PROPERTIES P2 & P3

Selecting K according to Eq. (11) assures the measure of relative sufficiency will indeed equal unity in the event that the statistic is classically sufficient; however, it has yet to be established that $S(\psi;\Theta,Z)$ is restricted to the interval $[0,1]$ and that $S(\psi;\Theta,Z)$ achieves zero when ψ is independent of θ. Although these results can be proven using Jensen's inequality and the convexity property of the logarithm, we can quickly achieve the same results by noting the connection between relative sufficiency and Shannon's information and employing some well known properties of information. Properties P2 and P3 follow directly from the following properties of mutual information. First, the mutual information between two random variables is zero iff they are statistically independent. Second, mutual information can never be negative. Third, processing can only destroy information. This last property implies that, because ψ is derived by a processing of Z (i.e. $\psi(z)$) without any direct link to θ, the mutual information between ψ and θ is guaranteed to be less that the mutual information between z and θ.

RELATIVE SUFFICIENCY AND STATE ESTIMATION

This section investigates relative sufficiency of subsets of measurements generated by a state variable model. Relative sufficiency is used to form a data segmentation criterion for a general state variable model.

THE STATE SPACE MODEL

Presented here is the state space model that is assumed responsible for the observed measurements. This modeling is composed of the standard state variable equations and statistical assumptions so often studied in the context of Kalman filtering theory. The relative sufficiency measure which we calculate is solely a function of these modeling assumptions and is only connected to Kalman filtering through this model. The segmentation criterion developed is a general result of these modeling assumptions and has nothing to do with any particular estimation scheme.

The time sequence of observations $z(k)$ is assumed to be generated by the following state space model.

$$x(k+1) = \Phi(k)x(k) + \Gamma(k)w(k) \tag{13a}$$

$$z(k+1) = H(k+1)x(k+1) + v(k+1) \tag{13b}$$

where:

$v(k); w(k)$: Measurement and Process noises are jointly white Gaussian noise processes such that

$$\begin{aligned} &\mathrm{E}\left[w(k)w'(k)\right] = Q(k) \\ &\mathrm{E}\left[w(k)w'(l)\right] = 0 \quad \forall k \neq l \\ &\mathrm{E}\left[v(k)v'(k)\right] = R(k) \\ &\mathrm{E}\left[v(k)v'(l)\right] = 0 \quad \forall k \neq l \\ &\mathrm{E}\left[w(k)v'(l)\right] = 0 \quad \forall k, l \end{aligned}$$

$x(0)$: The initial state and the process noise $w(k)$ are individually Gaussian and statistically independent (hence, also jointly Gaussian

This is a very general model that has a wide range of applicability to many practical problems. Thus, the data segmentation ideas illustrated here directly extend to other signal processing problems.

RELATIVE SUFFICIENCY OF RECENT OBSERVATION SETS

In this section we calculate the relative sufficiency of a simple class of statistics for generic Kalman filtering problems. We refer to this class of statistics as "data segmentation statisitics." Data segmentation statistics are best defined in terms of an example. Consider a measurement set whose most recent measurement is from time point k, i.e.,

$$z_L = \{z(k), z(k-1), z(k-2), \ldots, z(k-L)\} \tag{14}$$

where

$L+1$: The size of the complete original data set.

Then a data segmentation statistic is some subset of z_L

$$\psi(z) = z_s = \{z(i) | z(i) \in \text{some subset of z}\} \tag{15}$$

In particular, we will consider data segmentation statistics which restrict attention to the recent past history of the measurement set, i.e.,

$$\psi_l(z) = z_l = \{z(i)|i \in k, \ldots, k-l; \quad L \geq l \geq 1\} \tag{16}$$

Relative sufficiency is a measure of the degree to which this recent subset of measurements completely summarizes all the original data. Notice that we require at least one measurement for the sake of filter initialization. The oldest measurement is considered a form of apriori or minimal knowledge about the state at time k. The use of z(k-l) as a filter initializer will become apparent as we proceed.

We now develop a data segmentation criterion which specifies the number of past observations which are sufficient to estimate the current state. The functional form of the relative sufficiency will be in terms of the parameters of the previous state space model and the amount of past measurement history, l.

Motivated by the information-theoretic interpretation of the relative sufficiency measure, we set out to calculate the relative information between the data set, z_l or z_L, and the state at a time point k, $x(k)$. We denote such a state estimate by $\widehat{x}(k|z_l)$ or $\widehat{x}(k|z_L)$.

As defined by Shannon, the mutual information between two continuous random vector spaces, say X and Z, is defined in terms of the joint probability space of X and Z as

$$I(X;Z) = h(X) - h(X|Z) \tag{17}$$

where $h(X)$, the differential entropy of X, is given by

$$h(X) = -\mathrm{E}\left[\log\left[p_X(x)\right]\right] \tag{18}$$

$$= -\int_{\infty}^{\infty} p_X(x) \log\left[p_X(x)\right] dx, \tag{19}$$

and $h(X|Z)$, termed the differential entropy of X given Z, is defined as

$$h(X|Z) = -\mathrm{E}\left[\log\left[p_{X|Z}(x|z)\right]\right] \tag{20}$$

$$= -\int_{\infty}^{\infty} p_{X,Z}(x,z) \log\left[p_{X|Z}(x|z)\right] dx dz. \tag{21}$$

In terms of the differential entropy relative sufficiency is

$$S(Z_l; X, Z_L) = \frac{h(X(k) - h(X(k|z_l)}{h(X(k) - h(X(k)|z_L)}$$

From, Eqs. (19) and (21) it is clear that what is required is the apriori and conditional densities of the state $x(k)$: $p(x(k))$, $p(x(k)|z_L)$, $p(x(k)|z_l)$.

A Result From Information Theory If y is an n-dimensional normal random variable whose covariance matrix is P_y, then [7]

$$h(Y) = \frac{n}{2}\log\left[2\pi e |P_y|^{1/n}\right] \tag{22}$$

where $|P_y|$ is the determinant of the covariance of the normal vector y.

For $h(X(k))$ we assume (as previously discussed) a distribution based only on an initializing observation at time point k. Thus

$$h(X(k)) = \frac{1}{2}\log\left[|P_x(k|z(k))|\right] + \frac{n}{2}\log\left[2\pi e\right] \tag{23}$$

For the original data set z_L we have

$$h(X(k)|z(k), \ldots, z(k-L)) = \frac{1}{2}\log\left[|P_x(k|z(k), \ldots, z(k-L))|\right] + \frac{n}{2}\log\left[2\pi e\right] \tag{24}$$

Finally, for the segmented data set we have

$$h(X(k)|z(k), \ldots, z(k-l)) = \frac{1}{2}\log\left[|P_x(k|z(k), \ldots, z(k-l))|\right] + \frac{n}{2}\log\left[2\pi e\right] \tag{25}$$

Thus, it is clear from Eqs. (23), (24), and (25), that all we need to compute, in order to calculate the relative sufficiency of the data segmentation statistic, is the conditional covariances:

$$P_x(k|z(k))$$
$$P_x(k|z(k),\ldots,z(k-L))$$
$$P_x(k|z(k),\ldots,z(k-l))$$

These covariances are, of course, readily available from a Kalman filter or the corresponding matrix Riccati equation.

It follows that the measure of relative sufficiency for the data segmentation statistic is

$$\begin{aligned} S(Z_l;X,Z_L) &= \frac{h(X(k)-h(X(k|z_l)}{h(X(k)-h(X(k)|z_L)} \\ &= \frac{\log[|P_x(k|z(k))|]-\log[|P_x(k|z(k),\ldots,z(k-l))|]}{\log[|P_x(k|z(k))|]-\log[|P_x(k|z(k),\ldots,z(k-L))|]} \qquad (26) \\ &= \frac{\log\left[\frac{|P_x(k|z(k))|}{|P_x(k|z(k),\ldots,z(k-l))|}\right]}{\log\left[\frac{|P_x(k|z(k))|}{|P_x(k|z(k),\ldots,z(k-L))|}\right]} \qquad (27) \end{aligned}$$

Interpretation Of Relative Sufficiency
For The Gaussian Noise State Space Model The determinant of a covariance matrix of a Gaussian random variable is proportional to the volume of the hyper-ellipsoid traced out by the locus of constant probability. Hence, for the Gaussian case, the relative sufficiency, as expressed by Eq. (27), can be interpreted as the ratio of the statistic's ability to reduce the error volume to the ability of the original measurement set to reduce the error volume.

A TRACKING EXAMPLE

In this section, we take Singer's model[3] for maneuvering targets and derive the exact form of the relative sufficiency measure. This measure is then used to show that only a small subset of measurements representing the recent past are very nearly a sufficient statistic for all previous measurements. In effect, we provide a data segmentation criterion for Singer's target tracking problem.

Singer's model is briefly outlined. This model was used to develop a simulation which demonstrates the calculation of relative sufficiency. The goal is to show that relative sufficiency of the data segmentation statistic can be used as a criterion for the number of past measurements that need to be included in a Kalman filter tracker. A graph of the relative sufficiency is compared to graphs of the commonly used criterion (namely the settling to steady state of the diagonal elements of the error covariance) for comparison.

A Practical Issue Of Initialization Singer's model is used to define a Kalman tracking filter. Since our goal is to ultimately apply our results to a practical tracking problem, we must address the issue of filter initialization. In our tracking problem, two of the states are directly measurable (although, noise corrupted). This allows us to initialize the filter by setting these states equal to the values of the initializing measurement and to set the covariance of $\hat{x}_{\text{initial}}$ equal to a diagonal matrix with measurement variances for the measured states and an apriori choice of the variance for the unmeasured state. Thus, for the unconditional apriori density $p(x(k))$, we actually use the density $p(x(k)|z(k))$. Similarly, for the entire data set we need to compute $p(x(k)|z(k),\ldots,z(k-L))$ and for the segmenting statistic we need to compute $p(x(k)|z(k),\ldots,z(k-l))$. It is well known that, under the conditions of Eq. (13), the density of the state (both unconditionally and conditionally) is Gaussian [6].

SINGER'S MANEUVERING TARGET MODEL

Singer cast the problem of tracking a maneuvering target into the state equation model of Eq. (13). The exact details are available in [3]; but, briefly, his results when applied to a sensor which measures range and range-rate may be summarized as (actually, the following represents some practical approximations to Singer's derived model):

$$x = \begin{bmatrix} r = \text{range} \\ \dot{r} = \text{range-rate} \\ a = \text{radial acceleration} \end{bmatrix} \qquad (28)$$

$$\Phi = \begin{bmatrix} 1 & \Delta T & \frac{1}{2}\Delta T^2 \\ 0 & 1 & \Delta T(1 - \frac{1}{2}\beta) \\ 0 & 0 & \rho \end{bmatrix} \tag{29}$$

where

$$\rho = e^{-\beta \Delta T}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{30}$$

$$\Gamma Q \Gamma' = 2\beta\sigma_m^2 \begin{bmatrix} \Delta T^5/20 & \Delta T^4/8 & \Delta T^3/6 \\ \Delta T^4/8 & \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^3/6 & \Delta T^2/2 & \Delta T \end{bmatrix} \tag{31}$$

$$R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\dot r}^2 \end{bmatrix} \tag{32}$$

The parameters in this model have the following interpretation:

ΔT: time since last update (i.e. transition time)
β: 1/(maneuver time constant)
σ_r^2: range measurement variance
$\sigma_{\dot r}^2$: range-rate measurement variance
σ_m^2: maneuver variance

Reasonable values for these parameters for fighter aircraft and a typical radar sensor are:

$$\beta = .2(1/\text{sec.})$$
$$\sigma_r^2 = (200\text{ft.})^2$$
$$\sigma_{\dot r}^2 = (50\text{ft./sec.})^2$$
$$\sigma_m^2 = (1\text{g})^2 \qquad \text{g} = 32\text{ft./sec.}^2$$

Actually, in a real system, the measurement variances are a function of the signal-to-noise ratio, which is in turn a function of range, and hence not stationary. For the relatively short periods of time, however, the stationary assumption is not unreasonable.

SIMULATION RESULTS

Singer's model was coded along with the equations for computing relative sufficiency in Eq. (27). A regular sampling time of 1 sec. was assumed. The size $L+1$ of the original data set was taken to be large enough for the filters to reach steady state. This is equivalent to assuming the original data set reaches infinitely far back in time. As mentioned previously, we assume a segmented data set that is no smaller than one measurement at time k for initialization purposes. Thus, we have computed the relative sufficiency of: having only this one initializing measurement $z(k)$, having one additional measurement $z(k-1)$, having two additional measurements $z(k-1)$ and $z(k-2)$, etc.

Figure 1 shows a plot of the relative sufficiency of a segmented data set having from 1 to 20 additional measurements. Figures 2, 3, and 4 show the attainable error variances for the three states also as a function of the additional measurements utilized.

We see from Figure 1 that, for this particular tracking problem, eight or more of the most recent measurements are very nearly a sufficient statistic for an infinite measurement set. This is also supported by the transition to steady state of the diagonal terms of the error covariance matrix as depicted in Figures 2, 3, and 4. In light of the interpretation of error-volume reducing-power, Figure 1 shows that even as few as five of the most recent measurements retain 90% of the error-reducing capability of the original data set.

CONCLUSIONS

We have set up an axiomatic base upon which to build a measure of relative sufficiency. This base consists only of what one would reasonably desire of a measure of relative sufficiency. In a sense, the axiomatic base constitutes a definition of what one means by the term *relative sufficiency*. We have

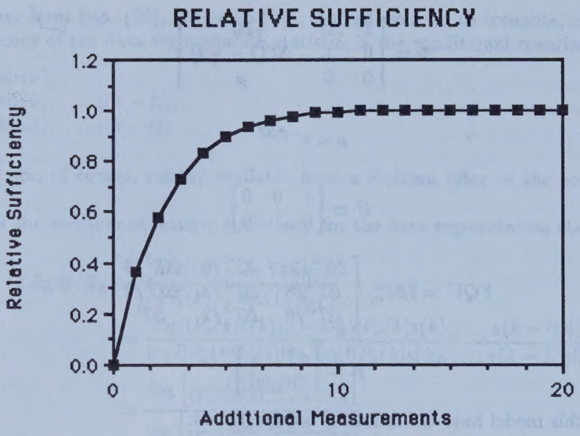


Figure 1: *Relative Sufficiency*

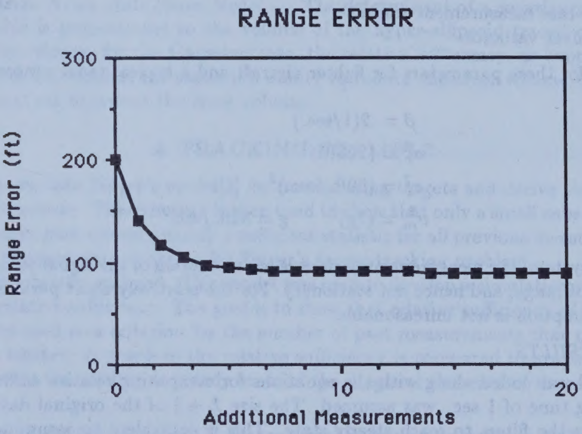


Figure 2: *Range Standard Deviation*

proven that this axiomatic base is in fact a definition, since it uniquely determines the functional nature of the measure of relative sufficiency.

Our measure of relative sufficiency can be successfully employed to investigate data segmentation statistics for standard tracking models. This measure provides a criterion for justifying the throwing away of older measurements in terms of the remaining measurements' ability to reduce the state error-variance. Although it is intuitively obvious that, for tracking, the error-covariance should in some way indicate the sufficiency of measurements, without relative sufficiency the exact usage of the error-covariance is not clear. In the practice of tracking, the diagonal terms of the covariance matrix are often used as indicators of the impact of suboptimal measurement handling (e.g., data segmentation). Our measure of relative sufficiency has the feature of utilizing the entire error covariance to produce a single scalar indicator.

ACKNOWLEDGEMENT

The work reported on in this paper was performed at the University of Southern California, Los Angeles, under National Science Foundation Grant ECS-8602531.

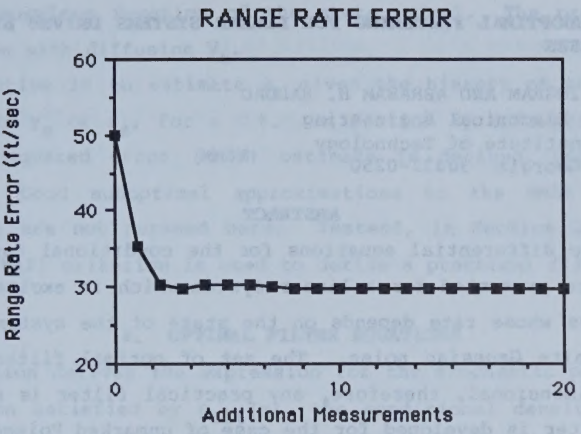


Figure 3: *Range-Rate Standard Deviation*

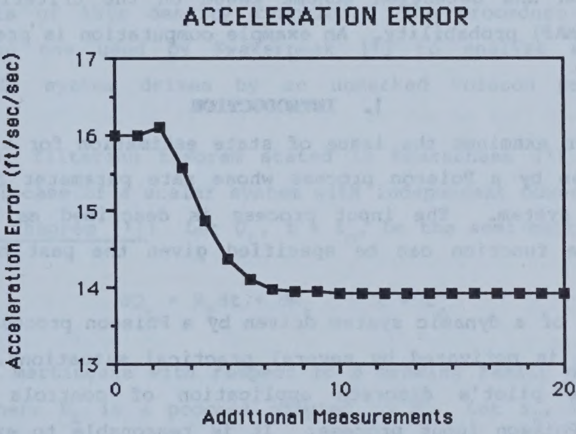


Figure 4: *Acceleration Standard Deviation*

REFERENCES

[1] H.W. Sorenson, *Parameter Estimation,* New York: Marcel Dekker Inc., 1980.

[2] R.A. Fisher, "On the mathematical foundations of theoretical statistics," *Philos. Trans. Roy. Soc. London, ser. A,* Vol. 222, 1921.

[3] R.A. Singer, "Estimating Optimal Tracking Filter Perfomrance for Manned Maneuvering Targets," *IEEE Trans. on Aerospace and Electronic Systems,* Vol. AES-6, No. 4, July 1970.

[4] R.G. Bartle, *The Elements of Real Analysis,* New York: John Wiley & Sons, 1976, p146.

[5] S. Kullback and R.A. Leibler, "On Information and Sufficiency," *Ann. Math. Statist.,* Vol. 22, 1951.

[6] J.M. Mendel *Lessons in Digital Estimation Theory,* New Jersey: Prentice-Hall, Inc., 1987.

[7] R.J. McEliece, *The Theory of Information and Coding,* London: Addison-Wesley Publishing Company, 1977 p44.

OPTIMAL AND SUBOPTIMAL FILTERING FOR LINEAR SYSTEMS DRIVEN BY SELF-EXCITED POISSON PROCESSES

MARY ANN INGRAM AND ABRAHAM H. HADDAD
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

ABSTRACT

Stochastic differential equations for the conditional density function and moments are presented for a linear system which is excited by a marked Poisson process whose rate depends on the state of the system and which is observed in white Gaussian noise. The set of optimal filtering equations is infinite dimensional, therefore, any practical filter is suboptimal. A suboptimal filter is developed for the case of unmarked Poisson excitation. This suboptimal filter estimates the Poisson process via a combined sequential estimation and detection scheme based on the criterion of maximum a posteriori (MAP) probability. An example computation is presented.

1. INTRODUCTION

This paper examines the issue of state estimation for a linear system which is driven by a Poisson process whose rate parameter depends on the state of the system. The input process is described as "self-excited" since its rate function can be specified given the past history of the input process.

The model of a dynamic system driven by a Poisson process with a state dependent rate is motivated by several practical situations. In aircraft maneuvers, the pilot's discrete application of controls is sometimes modeled as a Poisson input process. It is reasonable to expect that the rate of the control actions is dependent on the state of the aircraft. Another example is the tracking of a light source with a photon detector. The rate of photon arrivals certainly depends on the state of the tracking system, notably the tracking error angle.

The most general system considered in this paper is described by the following scalar equations:

$$dx_t = a_t x_t dt + b_t d\eta_t \tag{1}$$

$$y_t = \frac{dz_t}{dt} = c_t x_t + \frac{dw_t}{dt} \tag{2}$$

where η_t is a marked Poisson process whose marks (i.e., the amplitudes of the jumps) $\{u_i\}$ are a sequence of mutually independent, identically distributed random variables with density $p_u(u)$. The incident rate of η_t is a memoryless function of the state, $\mu(x_t)$. The process w_t is a Brownian motion with diffusion V_t.

The objective is to estimate x_t given the history of the observation process, either y_s or z_s, for $s < t$. In Section 2, an expression for the minimium mean-squared error (MMSE) estimate is derived, and shown to be impractical. Good suboptimal approximations to the MMSE estimate are desirable, but are not pursued here. Instead, in Section 3, the maximum a posteriori (MAP) criterion is used to derive a practical filter for x_t.

2. OPTIMAL FILTER EQUATIONS

This section derives the expression for the stochastic partial differential equation satisfied by $p_{t|t}(x)$, the conditional density function of x_t given $Z_t \overset{\Delta}{=} \{z_s; s < t\}$, based on a filtering theorem for white Gaussian observation noise. Furthermore, recursive equations are obtained for the central moments of this density function. The procedure used here is similar to the one used by Kwakernaak [1] to analyze a linear time invariant (LTI) system driven by an unmarked Poisson process with a constant rate.

First, the filtering theorem stated in Kwakernaak [1] is summarized for the special case of a scalar system with independent observation noise.

Filtering Theorem [1]: Let Q_t, $t > t_o$, be the semi-martingale defined by

$$dQ_t = R_t dt + dM_t \qquad t > t_o \tag{3}$$

where M_t is a martingale with respect to a growing family of σ-fields F_t, $t > t_o$, and where R_t is a process adapted to F. Let z_t, $t > t_o$, be the semi-martingale process

$$dz_t = h_t dt + dw_t \qquad t > t_o \tag{4}$$

where h is another process adapted to F, and w_t is a Brownian motion independent of F, such that $E(dw^2) = V_t dt$, $V_t > 0$ for $t > t_o$. Define Z_t as the growing family of σ-fields generated by the process z_t. For an arbitrary process ξ_t, define $\hat{\xi}_t \overset{\Delta}{=} E(\xi_t | Z_t)$. Then $\hat{Q}_t$ satisfies the dynamic equation

$$d\hat{Q}_t = \hat{R}_t dt + [\widehat{Q_t h_t} - \hat{Q}_t \hat{h}_t] V_t^{-1} [dz_t - \hat{h}_t dt] \tag{5}$$

The filtering theorem will be applied to $Q_t = e^{i\nu x_t}$, for x_t as defined in (1). However, the differential rule for filtered Poisson processes must first be used to obtain dQ_t. The rule may be found in Snyder [2, p. 200], and is also a special case of the differential rule for discontinuous semimartingales [1,3].

Differential Rule [2]: For an appropriately smooth function $Q(x_t)$ and for x_t defined in (1), the rule is

$$dQ(x_t) = a_t x_t\left(\frac{\partial Q(x_t)}{\partial x_t}\right)dt + \int_U \left[Q(x_t+b_t u) - Q(x_t)\right]K(dt,du) \qquad (6)$$

where the last integral is a counting integral [2, p. 195], evaluated over the mark space U, with respect to the Poisson counting measure K(dt,du). K(Δt,A) is the number of jumps of η_t during the interval Δt with marks in the set $A \subseteq U$.

Equation (6) may be put in the form of (3) by letting

$$dM_t = \int_U \left[Q(x_t+b_t u) - Q(x_t)\right]\left[K(dt,du) - \mu(x_t)p_u(u)dtdu\right] \qquad (7)$$

and taking $R_t dt$ as the remainder. The substitution of R_t into (5) yields

$$d\widehat{e^{i\nu x_t}} = \left(i\nu a_t \widehat{x_t e^{i\nu x_t}} + \widehat{e^{i\nu x_t}\left[e^{i\nu b_t u} - 1\right]\mu(x_t)}\right)dt$$

$$+ \left[\widehat{c_t x_t e^{i\nu x_t}} - c\hat{x}_t \widehat{e^{j\nu x_t}}\right]V_t^{-1}\left[dz_t - c_t\hat{x}_t dt\right] . \qquad (8)$$

Let $\theta_t = b_t u$ (recall u is the mark variable) and $p_{\theta_t}(\cdot)$ be the probability density function for θ_t. If it is assumed that the conditional density function $p_{t|t}(x)$ exists, then taking the inverse Fourier transform of each term of (8) yields

$$dp_{t|t}(x) = Lp_{t|t}(x)dt + V_t^{-1}c_t[x-\hat{x}_t]p_{t|t}(x)\left[dz_t - c_t\hat{x}_t dt\right] \qquad (9)$$

where L is the linear operator given by

$$Lp(x) = -\frac{\partial}{\partial x}\left[a_t x p(x)\right] + \left(p_{\theta_t}(x) * \left[\mu(x)p(x)\right]\right) - \mu(x)p(x) \qquad (10)$$

where "*" denotes convolution. As in Kwakernaak's case, equation (9) is the same as the Kushner equation for systems driven by Brownian motion, except for the definition of L.

Equations (9) and (10) can be used to derive stochastic differential equations for $\hat{x}_t$ and the n^{th} conditional central moments $P_{n,t} = E[(x_t-\hat{x}_t)^n|Z_t]$ as follows:

$$P_{n,t} = E[(x_t-\hat{x}_t)^n|Z_t] \qquad n = 1,2,\ldots \tag{11}$$

$$d\hat{x}_t = a_t\hat{x}_t dt + b_t E(u)\widehat{\mu(x_t)}dt + V_t^{-1}c_tP_{2,t}[dz_t - c_t\hat{x}_t dt]$$

$$\begin{aligned} dP_{n,t} = {} & na_tP_{n,t}dt + \sum_{k=1}^{n}\binom{n}{k}b_t^kE(u^k)\widehat{(x_t-\hat{x}_t)^{n-k}\mu(x_t)}dt - nb_tE(u)\widehat{\mu(x_t)}P_{n-1,t}dt \\ & + V_t^{-1}c_t[P_{n+1,t} - nP_{2,t}P_{n-1,t}][dz_t - c_t\hat{x}_t dt] \\ & + nV_t^{-1}c_t^2P_{2,t}\left[\frac{n-1}{2}P_{2,t}P_{n-2,t} - P_{n,t}\right]dt \qquad n = 2,3,\ldots \end{aligned} \tag{12}$$

Equations (11) and (12) represent an infinite set of coupled stochastic differential equations. Thus, an exact mean-squared error optimal filter is impossible to implement. Furthermore, in Kwakernaak's opinion, simple truncation of the moment equations (for the constant rate case) leads to unstable filters and generally poor results. Hence, approximate suboptimal filtering techniques are required, and are under investigation. This paper considers an alternative approach which uses a different error criterion, and is treated in the next section.

3. A MAP APPROACH

For this analysis, it is assumed that the driving process η_t is a counting process, i.e., it has only unit jumps. Furthermore, it is assumed that the system being driven is linear time-invariant, that is, $a_t = a$ and $b_t = b$ in equation (1). Thus, it is clear that knowledge of the jump times implies knowledge of x_t. The approach followed in this section is to obtain MAP estimates of the number N_T of jumps in η_t and the jump times $\underline{\tau}_{N_T} = [\tau_1,\tau_2,\ldots,\tau_{N_T}]$ on the interval [0,T), given the observations $Y_T =$

$\{y_s; s \leq T\}$. The state estimate at time T, denoted $\tilde{x}_T$, is then constructed by the appropriate superposition of impulse responses. The approach is made into a practical sequential algorithm by using time discretization and a finite time window.

This is an extension of the work of Au and Haddad [3] wherein the approach outlined above was taken for marked Poisson driving processes which have constant known rates.

The MAP estimates $\tilde{N}_T$ and $\tilde{\underline{\tau}}_M$ satisfy

$$(\tilde{N}_T, \tilde{\underline{\tau}}_M) = \arg\left(\max_{\substack{0 \leq N^* \leq M \\ \underline{\tau}^*_M \varepsilon R^M_+}} \left[\ell n\, p_{N_T, \underline{\tau}_M | Y_T, N_T \leq M}(N^*, \underline{\tau}^*_M) \right] \right) \tag{13}$$

where the argument of the logarithm is a joint a posteriori probability density function. M is an integer chosen large enough so that $\Pr[N_T > M]$ is negligible. The condition $N_T \leq M$ ensures that $\underline{\tau}_M$ includes enough jump times to construct $\tilde{x}_T$.

The log of the density function in (13) can be replaced by the following expression without changing the result:

$$\ell n \frac{1}{2V_T} \int_0^T \left[2y_t - \sum_{i=0}^{N^*} h(t, \tau^*_i)\right]\left[\sum_{j=0}^{N^*} h(t, \tau^*_j)\right] dt$$

$$+ \ell n\, p_{\underline{\tau}_M | N_T, N_T \leq M}\left(\underline{\tau}^*_M | N_T = N^*, N_T \leq M\right) + \ell n \Pr\left[N_T = N^* | N_T \leq M\right] . \tag{14}$$

The first term is recognized as the log likelihood function, wherein $h(t, \tau^*_j)$ represents the response of the system at time t to an impulse at time τ^*_j. For brevity, $h(t, \tau^*_o)$ is defined as the unforced response due to a known initial condition x_o.

The next objective is to simplify the expressions of the second and third terms of (14). Note that the event $N_T = N^*$ is also the event $\tau_{N^*} \leq T < \tau_{N^*+1}$. Therefore, the probability density in the second term can be rewritten as

$$p_{\underline{\tau}_M | N_T, N_T \leq M}\left(\underline{\tau}^*_M | N_T = N^*, N_T \leq M\right) =$$

$$\begin{cases} \dfrac{p_{\underline{\tau}_M | N_T \leq M}\left(\underline{\tau}^*_M | N_T \leq M\right)}{\Pr\left[\tau_{N^*} \leq T, \tau_{N^*+1} > T | N^* \leq M\right]} & \text{for } 0 \leq \tau^*_1 \leq \ldots \leq \tau^*_{N^*} \leq T \\ & \text{and } T < \tau^*_{N^*+1} \leq \tau^*_M \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Since $\ell n\, 0 = -\infty$, it is reasonable to restrict the region over which the expression in (13) is maximized to the region of support of (15). Under this restriction, the third term of (14) cancels the denominator of the nonzero part of (15).

The remaining term to simplify is the numerator of the nonzero part of (15). It is noted that the event $N_T < M$ is also the event $\tau_{M+1} > T$. Thus, the term of interest may be expressed as a marginal density function:

$$p_{\underline{\tau}_M | N_T < M}\left(\underline{\tau}^*_M | N_T < M\right) = \int_T^{\infty} p_{\underline{\tau}_{M+1} | \tau_{M+1} > T}\left(\underline{\tau}^*_{M+1} | \tau_{M+1} > T\right) d\tau^*_{M+1} \ . \tag{16}$$

It is noted that the region of support of the integrand is over the "wedge" $0 < \tau_1 < \dots < \tau_M < \tau_{M+1}$ minus the half space $\tau_{M+1} < T$. Therefore, (16) can be rewritten as:

$$p_{\underline{\tau}_M | N_T < M}\left(\underline{\tau}^*_M | N_T < M\right) = \int_{\mathrm{Max}\left(\tau^*_M, T\right)}^{\infty} \frac{p_{\underline{\tau}_{M+1}}\left(\underline{\tau}^*_{M+1}\right)}{\Pr\left(\tau_{M+1} > T\right)} d\tau^*_{M+1} \tag{17}$$

The unconditional density in the integrand of (17) is a special case of the density considered by Snyder [2, p. 248] for a self-exciting point process. For this special case, the density can be expressed as:

$$p_{\underline{\tau}_{M+1}}\left(\underline{\tau}^*_{M+1}\right) = \begin{cases} \displaystyle\prod_{i=1}^{M+1} \frac{\partial}{\partial \tau^*_i} \ -\exp \int_{\tau^*_{i-1}}^{\tau^*_i} -\mu\left[\overline{x}_t\left(\underline{\tau}^*_{i-1}\right)\right] dt & \text{for } 0 < \tau^*_1 < \dots < \tau^*_{M+1} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

with

$$\overline{x}_t\left(\underline{\tau}^*_{i-1}\right) = x_o e^{at} u_1(t) + \sum_{j=1}^{i-1} b \exp\left[a\left(t - \tau^*_j\right)\right] u_1\left(t - \tau^*_j\right)$$

where u_1 is the unit step function. In words, $\overline{x}_t\left(\underline{\tau}^*_{i-1}\right)$ is the value of the state assuming that η_t has had jumps only at times $\tau^*_1, \dots, \tau^*_{i-1}$. Let $\overline{x}_t\left(\underline{\tau}^*_o\right)$ denote the unforced value of the state.

Substitution of (18) into (17) is straightforward due to the product form of (18) and yields

$$p_{\underline{\tau}_M | N_T < M}\left(\underline{\tau}^*_M | N_T < M\right) = \frac{p_{\underline{\tau}_M}\left(\underline{\tau}^*_M\right)}{\Pr\left(\tau_M > T\right)} \exp \int_{\tau^*_M}^{\mathrm{Max}\left(\tau^*_M, T\right)} -\mu\left[\overline{x}_t\left(\underline{\tau}^*_M\right)\right] dt \tag{19}$$

where it has been assumed that there exists some $\alpha > 0$ such that $\mu[x] > \alpha$ for all x, thus making

$$\exp \int_{\tau^*_M}^{\infty} - \mu[\overline{x}_t] dt = 0 \ .$$

It is noted that $p_{\underline{\tau}_M}(\cdot)$ is defined by replacing M + 1 by M in (18). Evaluation of the derivatives yields:

$$p_{\underline{\tau}_M}(\underline{\tau}^*_M) = \begin{cases} \prod_{i=1}^{M} \mu[\overline{x}_{\tau^*_i}(\underline{\tau}^*_{i-1})] \exp \int_{\tau^*_{i-1}}^{\tau^*_i} - \mu[\overline{x}_t(\underline{\tau}^*_{i-1})]dt & \text{for } 0 < \tau^*_1 < \dots < \tau^*_M \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The combination of equations (14), (15), (19), and (20) results in a new MAP equation:

$$(\tilde{N}_T, \underline{\tilde{\tau}}_M) =$$

$$\arg \left\{ \underset{0 < N^* < M}{\text{Max}} \left[\underset{\substack{\underline{\tau}^* \varepsilon R^M_+ \\ \tau^*_1 < \dots < \tau^*_{N^*} < T \\ T < \tau^*_{N^*+1} < \dots < \tau^*_M}}{\text{Max}} \frac{1}{2V_T} \int_0^T [2y_t - \overline{x}_t(\underline{\tau}^*_M)][\overline{x}_t(\underline{\tau}^*_M)]dt + \ell n \left(\prod_{i=1}^{M} \mu[\overline{x}_{\tau^*_i}(\underline{\tau}^*_{i-1})] \right) - \int_0^{\text{Max}(\tau^*_M, T)} \mu[\overline{x}_t(\underline{\tau}^*_M)]dt \right] \right\} \tag{21}$$

where the maximization is to be performed in two steps, first over the $\underline{\tau}^*_M$'s for fixed N*, and second over the N*'s.

4. SEQUENTIAL MAP APPROXIMATION

The MAP equation (21) derived in the previous section is now approximated as a sequential algorithm. In this approximation, the observations are processed in subintervals each of length Δ, which is chosen such that the probability of having two or more jumps in each interval is negligibly small. Each subinterval of observations is used to detect a jump in the subinterval and to estimate the jump time, as well as to update the estimates of past jump times.

In order to reduce computational complexity of the algorithm, estimates further than L subintervals away from the new subinterval are not updated and considered "finalized." The selection of L represents a tradeoff between performance and complexity. Thus, observations in the k^{th} subinterval $[(K-1)\Delta, K\Delta)$ are used to update estimates in the "window" $[(K-L)\Delta, K\Delta)$. $\tilde{N}_{(K-L)\Delta}$ represents the number of finalized estimates of jump times.

Equation (21) is next modified so that maximization is performed only over jump times occurring after the time (K-L)Δ. Any additive terms which depend solely on finalized estimates are dropped. For brevity, let $\tilde{N}_F$ =

$\tilde{N}_{(K-L)\Delta}$ ("F" for finalized). Furthermore, redefine $\underline{\tau}_L$ as $\left[\tau_{\tilde{N}_F+1},\ldots,\tau_{\tilde{N}_F+L}\right]$, and redefine $\bar{x}_t\left(\underline{\tau}^*_L\right)$ as the state assuming that jumps have occurred only at the finalized times and at the proposed times $\underline{\tau}^*_L$. The modified (approximate) version of (21) is:

$$\left(\tilde{N}_{K\Delta},\underline{\tilde{\tau}}_L\right) \approx \arg \left\{ \underset{\tilde{N}_F < N^* < \tilde{N}_F + L}{\text{Max}} \left[\underset{\substack{\tau_{\tilde{N}_F} < \tau^*_{\tilde{N}_F+1} < \ldots < \tau^*_{N^*} < K\Delta \\ K\Delta < \tau^*_{N^*+1} < \ldots < \tau^*_{\tilde{N}_F+L}}}{\text{Max}} \right.\right.$$

$$\frac{1}{2V_T} \int_{(K-L)\Delta}^{K\Delta} \left[2y_t - \bar{x}\left(\underline{\tau}^*_L\right)\right]\left[\bar{x}\left(\tau^*_L\right)\right]dt$$

$$\left.\left. + \ln\left(\int_{i=\tilde{N}_F+1}^{\tilde{N}_F+L} \mu\left[\bar{x}_{\tau^*_i}\left(\underline{\tau}^*_{i-1}\right)\right]\right) - \int_{(K-L)\Delta}^{\text{Max}\left(\tau^*_{\tilde{N}_F+L},K\Delta\right)} \mu\left[\bar{x}_t\left(\underline{\tau}^*_L\right)\right]dt \right] \right\} \qquad (22)$$

There is a remaining difficulty with the maximization over the τ^*'s in (22). Assume that this maximization is being performed for a given, fixed N^*. Furthermore, assume a discretized domain, i.e., a subset of equally spaced discrete values in R^L. The discretization implies that the expression in (22) is evaluated over a finite number of values for the τ^*'s between $\tau_{\tilde{N}_F}$ and $K\Delta$, but there are still an infinite number of values to check for the τ^*'s above $K\Delta$. Maximizing over these "future" jump times is equivalent to maximizing the joint a priori probability density for these jump times.

The constant rate case $\left(\mu[x_\varepsilon] = \mu_o\right)$ presents no difficulty, because the joint a priori density function for the jump times after $K\Delta$ has its maximum at $\tau^*_{N^*+1} = \tau^*_{N^*+2} = \ldots = \tau^*_{\tilde{N}_F+L} = K\Delta$. It is easily shown that the same is true for stable first order systems and rate functions $\mu[x]$ which monotonically increase with $|x|$. However, for more general LTI systems and rate functions, finding the maximum of the a priori joint density is apparently not as easy. This matter is currently under investigation.

5. EXAMPLE

Figures 1 and 2 display simulation results based on the algorithm of Section 4. The parameters are (see equations 1 and 2) $a_t = -5$, $b_t = 2$, and $c_t = 1$. The rate or intensity, $\mu(x_t)$, of the counting process η_t, takes only two values: $\mu(x_t) = 2$ for $|x_t| < 1$ and $\mu(x_t) = 4$ for $|x_t| > 1$. Figure 1 contains the state trajectory. The rate takes its high value when the trajectory is above the dashed line and the low value otherwise.

For estimation, $\Delta = 0.03125$ sec. This yields an approximate upper bound for $\Pr[\eta_{t+\Delta} - \eta_t \geq 1]$ of $4\Delta = 0.125$. The observation noise samples have a standard deviation $(\sqrt{V_t})$ of 0.15. The estimation/detection window is $L = 4$. Estimation results are shown in figure 2. Some errors may be observed at $t \cong 2$ and $3 < t < 4$. It is noted that for $\sqrt{V_T} = 0.1$, all of the jumps were correctly detected (to the order of the simulation sample period) and for $\sqrt{V_T} = 0.2$, several more false detections occurred in the region $0.5 < t < 1.5$.

6. CONCLUSIONS

The state estimation problem has been considered for a linear system observed in additive white Gaussian noise, where the system is driven by a Poisson process with a state dependent rate. It is no surprise that the minimum mean-squared estimator is infinite dimensional, since the same is true for the simpler constant rate case. However, it is expected that the form of the equations will suggest a good suboptimal approximation in the future. An implementable estimator was developed based on maximum a posteriori (MAP) estimates of the number and times of the jumps in the driving process. However, the feasibility of this scheme has been shown only for certain LTI systems and rate functions. Further investigation is needed to enlarge the apparently limited applicability of this MAP approach.

ACKNOWLEDGEMENTS

This research is supported by the U.S. Air Force under Contracts F08635-84-C-0273 with the Armament Laboratory and AFOSR-87-0308.

REFERENCES

[1] H. Kwakernaak, "Filtering for systems excited by Poisson white noise," Eds.: A. Bensoussan, J. L. Lions, Control Theory, Numerical Methods, and Computer Systems, Springer Lecture Notes in Economics and Mathematical Systems, vol. 107, Berlin, pp. 468-492, 1975.

[2] D. L. Snyder, Random Point Processes, Wiley, New York, 1975.

[3] S. P. Au and A. H. Haddad, "Suboptimal sequential estimation-detection scheme for Poisson driven linear systems," Inform. Sci., vol. 16, pp. 95-113, 1978.

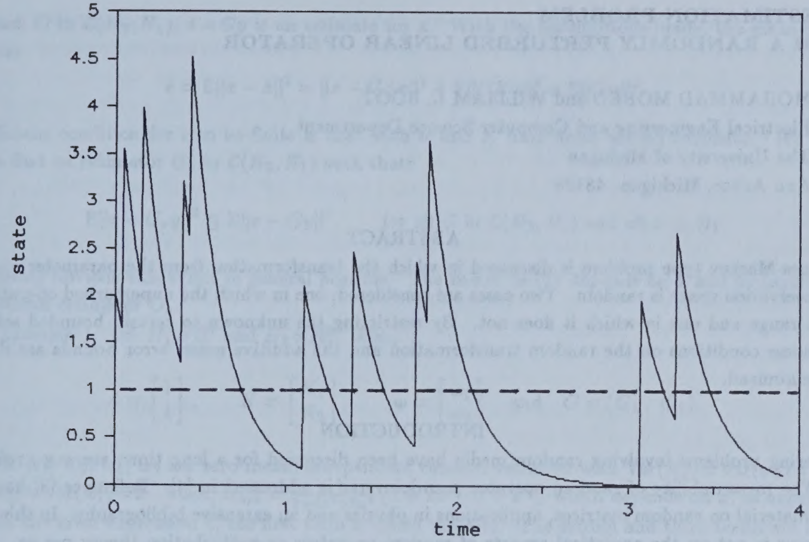


Figure 1. Example state trajectory

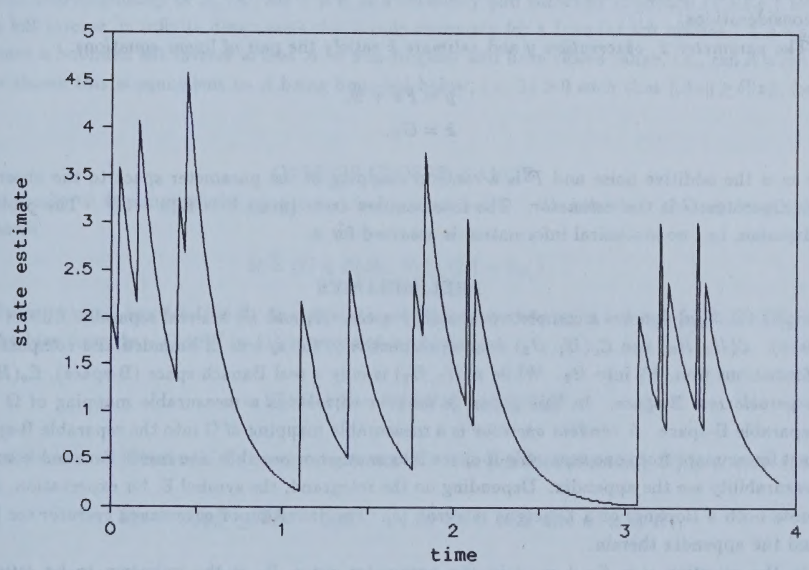


Figure 2. Estimate output

AN ESTIMATION PROBLEM WITH A RANDOMLY PERTURBED LINEAR OPERATOR

MOHAMMAD MOBED and WILLIAM L. ROOT
Electrical Engineering and Computer Science Department
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

A Gauss-Markov type problem is discussed in which the transformation from the parameter space to the observation space is random. Two cases are considered; one in which the unperturbed operator has closed range and one in which it does not. By restricting the unknown to certain bounded sets and with some conditions on the random transformation and the additive noise, error bounds are derived and minimized.

INTRODUCTION

Scattering problems involving random media have been discussed for a long time; see, e.g., reference [9]. The measurability of random operator pseudoinverse is addressed in [4]. Reference [8] has some good material on random matrices, applications in physics and an extensive bibliography. In this paper the focus is not on the analytical aspects of random operators or perturbation theory per se. There do arise some questions of more or less theoretical nature, which are commented upon mainly in the appendix. Those applications in which the spaces involved are finite dimensional, are not affected by such considerations.

The *parameter* x, *observation* y and *estimate* $\hat{x}$ satisfy the pair of linear equations:

$$y = Fx + w,$$
$$\hat{x} = Gy,$$

where w is the additive noise and F is a random mapping of the parameter space to the observation space. Operator G is the *estimator*. The mean-square error (m.s.e.) is $\mathrm{E}\|x - \hat{x}\|^2$. The problem is non-Bayesian, i.e., no statistical information is assumed for x.

PRELIMINARIES

The triplet $(\Omega, \mathcal{F}, \mu)$ denotes a complete probability space. H_1 and H_2 are real separable Hilbert spaces (H-space). $\mathcal{L}(H_1, H_2)$ and $\mathcal{L}_o(H_1, H_2)$ denote, respectively, the spaces of bounded and compact linear transformations from H_1 into H_2. While $\mathcal{L}(H_1, H_2)$ is only a real Banach space (B-space), $\mathcal{L}_o(H_1, H_2)$ is a *separable* real B-space. In this paper, a *random variable* is a measurable mapping of Ω into a real separable H-space. A *random operator* is a measurable mapping of Ω into the separable B-space of compact linear maps from one separable H-space into another or possibly into itself. For some comments on measurability see the appendix. Depending on the integrand, the symbol E, for expectation, is used to denote both a Bochner or a Lebesgue integral, [3]. For definition of *covariance operator* see [1], pp 421 and the appendix therein.

In the equation $y = Fx + w$, x in the parameter space H_1 is the unknown to be estimated. Also, $F = A + K$, where A in $\mathcal{L}(H_1, H_2)$ is a fixed known mapping; the so-called unperturbed or the average operator. Throughout it is assumed A has zero kernel ($\ker A = 0$). The H_2-valued random variable w has zero mean, i.e., $\mathrm{E}w = \int_\Omega w(\omega)\, d\mu(\omega) = 0_{H_2}$. For the case of closed range, when both H_1 and H_2 are infinite dimensional, we will have to assume w has finite second moment, i.e., $\mathrm{E}\|w\|^2 = \int_\Omega \|w(\omega)\|^2\, d\mu(\omega) < \infty$. The $\mathcal{L}_o(H_1, H_2)$-valued random operator K is also assumed to have zero mean: $\mathrm{E}K = \int_\Omega K(\omega)\, d\mu(\omega) = 0_{\mathcal{L}_o(H_1,H_2)}$. We say K has finite second moment if $\mathrm{E}\|K\|^2 = \int_\Omega \|K(\omega)\|^2\, d\mu(\omega) < \infty$ where $\|K(\omega)\|$ is the operator norm of $K(\omega)$. It is always assumed that K and w are independent, i.e., $\mu(w^{-1}(B_1) \cap K^{-1}(B_2)) = \mu(w^{-1}(B_1))\mu(K^{-1}(B_2))$ for all Borel sets B_1 of H_2 and B_2 of $\mathcal{L}_o(H_1, H_2)$.

For each G in $\mathcal{L}(H_2, H_1)$, $\hat{x} = Gy$ is an estimate for x. With the assumptions made, the mean-square error is:

$$e = \mathrm{E}\|x - \hat{x}\|^2 = \|x - GAx\|^2 + \mathrm{E}\|GKx\|^2 + \mathrm{E}\|Gw\|^2. \tag{1}$$

A sufficient condition for e to be finite is that both w and K have finite second moments. We would like to find an estimator G_o in $\mathcal{L}(H_2, H_1)$ such that:

$$\mathrm{E}\|x - G_o y\|^2 \leq \mathrm{E}\|x - Gy\|^2, \qquad \text{for all } G \text{ in } \mathcal{L}(H_2, H_1) \text{ and all } x \text{ in } H_1. \tag{2}$$

It is pretty obvious this is not in general possible. The m.s.e. in (1) depends on x and so usually will the optimal estimator G_o.

Example 1. Let $H_1 = R^1$ and $H_2 = R^2$. Also,

$$A = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad \text{and} \quad G = [G_1 \quad G_2].$$

Assume K_1, K_2, w_1, w_2 are zero mean, independent random variables with $\mathrm{Var}(K_1) = \mathrm{Var}(w_2) = 1$ and $\mathrm{Var}(K_2) = \mathrm{Var}(w_1) = 2$. Then, $G_{\text{opt}} = \frac{x^2}{2+8x^2+5x^4}[1 + 2x^2 \quad 2 + x^2]$, which depends on x, as expected.

In the error expression 1, the first term is called the *bias*. The second and third terms are due to the "multiplicative" noise K and the additive noise w. An estimator is said to be *unbiased* if $\mathrm{E}\hat{x} = x$, $\forall x \in H_1$. This is equivalent to $GA = \mathrm{I}_{H_1}$, i.e., to say G is a left inverse of A. While in finite dimensional problems non-singularity of A, i.e., $\ker A \neq 0$, is a necessary and sufficient condition (n.a.s.c.) for A to have a left inverse, in infinite dimensions this is only necessary for a *bounded* left inverse. A n.a.s.c. for A to have a bounded left inverse is that A be non-singular and have closed range, i.e., $\operatorname{ran} A = \overline{\operatorname{ran} A}$. It can be shown this is equivalent to A being bounded below, i.e, $\exists\, \delta > 0$ such that $\|Az\| \geq \delta \|z\|$, for all z in H_1.

CASE OF CLOSED RANGE

In this section it is assumed the operator A is bounded below. We introduce the set $\mathcal{U}$ of unbiased estimators:

$$\mathcal{U} \overset{\mathrm{d}}{=} \{G \in \mathcal{L}(H_2, H_1) : GA = \mathrm{I}_{H_1}\}.$$

Closedness of $\operatorname{ran} A$ implies $\mathcal{U} \neq \emptyset$. In particular, the psuedoinverse of A belongs to $\mathcal{U}$; see [1]. For G in $\mathcal{U}$ the bias term $\|x - GAx\|^2$ in (1) is zero and we have:

$$e = \mathrm{E}\|x - \hat{x}\|^2 = \mathrm{E}\|GKx\|^2 + \mathrm{E}\|Gw\|^2, \tag{3}$$

which still has x in it through the random operator K. The problem of finding G_o in $\mathcal{U}$ such that:

$$\mathrm{E}\|x - G_o y\|^2 \leq \mathrm{E}\|x - Gy\|^2, \qquad \text{for all } G \text{ in } \mathcal{U} \text{ and all } x \text{ in } H_1, \tag{4}$$

in general, has no solution. Example 1 can be redone with the unbiasedness condition $G_1 + G_2 = 1$ added. The result is: $G_{\text{opt}} = \frac{1}{3(1+x^2)}[1 + 2x^2 \quad 2 + x^2]$.

Definition. By a ***suboptimal estimator*** is meant here a bounded linear mapping from the observation space into the parameter space, which minimizes an error upper bound, rather than the error itself.

The suboptimal estimators found for the case of closed range will be unbiased and will be denoted by G_o, though they are different from the G_o of inequality 4. Four bounds and corresponding estimators are obtained. Except for the first one which is curiously simple, they all have the form: $\operatorname{trace}[GCG^*]$, and are minimizable by the following lemma, given here without proof.

Lemma 1. (modified statement of theorem 2.2 of [1]) Let H_1 and H_2 be separable H-spaces and A in $\mathcal{L}(H_1, H_2)$ be bounded below. Also let C be a self-adjoint positive-definite trace-class operator on H_2, and $C^{-1/2} \overset{\mathrm{d}}{=} (C^{1/2})^{-1}$.

a. Assume $D = \{z \in H_1 : Az \in \operatorname{ran} C^{1/2}\}$ is dense in H_1. Define $B : H_1 \to H_2$ by $\operatorname{dom} B = D$, $B(z) = C^{-1/2}A(z)$, $\forall z \in D$. Let B^+ be the pseudoinverse of B in the sense of [1].

b. Assume $B^+C^{-1/2}$ is continuous. Let G_o be the continuous extension of $B^+C^{-1/2}$.

Then, the bounded operator $G : H_2 \to H_1$ minimizing trace$[GCG^*]$ and satisfying $GA = \mathrm{I}_{H_1}$ is given, uniquely by $G = G_o$.

Remark. The condition that C be trace class is used in the proof of theorem 2.2 of [1] only to guarantee that GCG^* has finite trace. If H_1 is finite dimensional, this condition can be dropped. If, in addition, C is onto, it can be shown that both hypotheses a. and b. are satisfied and G_o has a closed form given below. Things are even simpler when H_2 is finite dimensional; then H_1 is also finite dimensional and once again a. and b. are satisfied and we have a formula for G_o. Let us summarize all this:

Corollary. If H_1 is finite dimensional and C is onto, or, if H_2 is finite dimensional, then G_o of lemma 1 is given by:

$$G_o = (A^*C^{-1}A)^{-1}A^*C^{-1}, \tag{5}$$

which is the usual formula for finite dimensional problems.

Pseudoinverse Estimator.

It is possible to upper-bound the error in a straightforward manner, with no special hypotheses such as finite dimensionality of the spaces, by only requiring that w and K have finite second moments. From (3) it follows:

$$e \leq (\mathrm{E}\|Kx\|^2 + \mathrm{E}\|w\|^2)\|G\|^2. \tag{6}$$

So, $\|G\|$ is to be minimized subject to $GA = \mathrm{I}_{H_1}$. It can be shown that if G_o in $\mathcal{U}$ is defined by:

$$G_o(y) = \begin{cases} A_r^{-1}y, & \text{if } y \in \operatorname{ran} A; \\ 0, & \text{if } y \in (\operatorname{ran} A)^{\perp}; \end{cases}$$

where A_r^{-1} is the inverse of A defined on $\operatorname{ran} A$, then $\|G_o\| \leq \|G\|$ for all G in $\mathcal{U}$. So $G_o = A_r^{-1}P$, where P is the orthogonal projection of H_2 onto $\operatorname{ran} A$. According to theorem 1.2 of [1], pp 407, this is just the pseudoinverse of A which through corollary 1.5 of [1], pp 416, can be written in terms of the adjoint as follows:

$$G_o = A_r^{-1}P = A^+ = (A^*A)^{-1}A^*.$$

$$\text{minimum bound } = (\mathrm{E}\|Kx\|^2 + \mathrm{E}\|w\|^2)\|A^+\|^2,$$

$$\text{corresponding m.s.e.} = \mathrm{E}\|A^+Kx\|^2 + \mathrm{E}\|A^+w\|^2 = \operatorname{trace}[A_r^{-1}P(C_{Kx} + C_w)(A_r^{-1}P)^*],$$

where C_{Kx} and C_w are the covariance operators of random variables Kx and w. Use has been made of the fact that if u is a random variable and T is a bounded linear map, then $\mathrm{E}\|Tu\|^2 = \operatorname{trace}[TC_uT^*]$; [1], pp 421.

Norm Constraint I.

Here we suppose it is given that $\|x\| \leq r$, otherwise written as $x \in \mathrm{S}_r$, where S_r is the closed ball of radius r centered at the origin of H_1 and r is a known positive constant. In the theorem that follows we derive an upperbound for e, the mean-square error, subject to the condition that $\|x\| \leq r$ and, of course, assuming the estimator is unbiased.

The unbiasedness of G implies it must be onto, i.e., $\operatorname{ran} G = H_1$. On the other hand, we will dominate the operator norm of G by its Hilbert-Schmidt norm. The two conditions $\operatorname{ran} G = H_1$ and $\|G\|_{HS} < \infty$ are compatible if and only if H_1 is finite dimensional. Here is why: If G is Hilbert-Schmidt, then it is also compact. A closed linear manifold (in this case H_1), contained in the range of a compact operator (in this case G), is necessarily finite dimensional; [7], theorem 8, pp 19. This proves H_1 must be finite dimensional. Implication in the opposite direction is trivial.

Theorem 1. Assume H_1 is finite dimensional and $\|x\| \le r$. Then,

$$e = \mathrm{E}\|x - \hat{x}\|^2 \le \mathrm{trace}[GCG^*], \quad \text{for all } G \in \mathcal{U} \quad \text{and} \quad G_o = (A^*C^{-1}A)^{-1}A^*C^{-1},$$

where $C = r^2\mathrm{E}\|K\|^2\mathrm{I}_{H_2} + C_w$.

Proof:

$$\begin{aligned} e &= \mathrm{E}\|GKx\|^2 + \mathrm{E}\|Gw\|^2 \\ &\le \|G\|^2\mathrm{E}\|K\|^2\|x\|^2 + \mathrm{E}\|Gw\|^2 \\ &\le \|G\|^2_{HS}\mathrm{E}\|K\|^2\|x\|^2 + \mathrm{E}\|Gw\|^2 \\ &\le r^2\mathrm{E}\|K\|^2\mathrm{trace}[GG^*] + \mathrm{trace}[GC_wG^*] \\ &= \mathrm{trace}\, G[r^2\mathrm{E}\|K\|^2\mathrm{I}_{H_2} + C_w]G^*, \end{aligned}$$

where $\mathrm{E}\|K\|^2$ is a known constant and C_w is the covariance operator of w, also known. This concludes the proof of the inequality. To show C is onto, one approach would be the following: For any densely defined closed operator M, $I + M^*M$ has an everywhere defined bounded inverse and is therefore both one-to-one and onto; [5] pp 307. Now, $r^2\mathrm{E}\|K\|^2$ is a positive constant and plays no significant role, and C_w is a self-adjoint nonnegative-definite operator and can be written as $(C_w^{1/2})^*(C_w^{1/2})$ where $C_w^{1/2}$ is the unique self-adjoint nonnegative-definite square root of C_w. Of course, both C_w and $C_w^{1/2}$ are densely defined and closed since they are continuous operators on H_2. The above fact now applies to conclude that C is onto. Conditions of the corollary to lemma 1 are now met, namely $\dim H_1$ is finite and C is onto. So G_o is indeed given by the above formula. The minimum bound, i.e., $\mathrm{trace}[G_oCG_o^*]$, is given by $\mathrm{trace}[(A^*C^{-1}A)^{-1}]$. ∎

Norm Constraint II and the Parallelepiped Constraint.

The upper bound in theorem 1 is useless if H_1 is infinite dimensional. Here we shall derive bounds that do not involve $\|G\|_{HS}$ or dimensionality of the spaces. But we do need some conditions on the random operator K.

Definition. We shall say K is *uncorrelated* if its matrix has uncorrelated entries relative to some pair of orthonormal bases for H_1 and H_2.

Both the norm constraint $x \in \mathrm{S}_r$ and the parallelepiped constraint $x \in \Pi_b$ are considered in what follows, where Π_b is the compact parallelepiped centered at the origin of H_1 and with edges $b_1, b_2, \ldots$ where $b_i \ge 0$ and $\sum b_i^2 < \infty$:

$$\Pi_b = \{u \in H_1 : |u_i| \le b_i,\ i = 1, 2, \ldots\}.$$

While S_r does not depend on what basis is used for H_1, Π_b does. It is always assumed Π_b is aligned with respect to that basis of H_1, which together with some basis of H_2, render K uncorrelated. The details of getting the bounds for these two constraints are almost identical and the bounds themselves have exactly the same form. Also, the operator A does not enter the discussion at all; therefore these inequalities will be valid even when A does not have closed range and G is in $\mathcal{L}(H_2, H_1)$ and not necessarily in $\mathcal{U}$. In particular, lemma 4 will be used again when we look at the ill-posed problem.

Lemma 2. Suppose K is uncorrelated relative to $\mathcal{B}_1 = (e_i)_{i=1}^\infty$ and $\mathcal{B}_2 = (f_j)_{j=1}^\infty$, orthonormal bases of H_1 and H_2, respectively. Then the matrix of C_{Kx} is diagonal with respect to $\mathcal{B}_2$. (Since $\mathrm{E}\|Kx\|^2$ is finite, $\mathrm{trace}[C_{Kx}]$ is also finite.)

Proof: First of all, we note that Kx has zero mean; $\mathrm{E}(Kx) = (\mathrm{E}K)x = 0_{H_2}$. So, the (i,j)th entry of C_{Kx} is, by definition:

$$\begin{aligned} C_{Kx}(i,j) = \langle C_{Kx}f_j, f_i\rangle &= \mathrm{E}\langle Kx, f_i\rangle\langle Kx, f_j\rangle \\ &= \textstyle\sum_{m,n} x_m x_n \mathrm{E}\langle Ke_m, f_i\rangle\langle Ke_n, f_j\rangle \\ &= \textstyle\sum_{m,n} x_m x_n \sigma_{im}^2\, \delta_{ij}\delta_{mn} \\ &= \left(\textstyle\sum_n x_n^2\sigma_{in}^2\right)\delta_{ij}, \end{aligned}$$

where $\sigma_{in}^2 = \mathrm{Var}(K_{in})$, $K_{in} = \langle Ke_n, f_i \rangle$ and $x_n = \langle x, e_n \rangle$. ■

Lemma 3. Suppose trace-class operators C_1 and C_2 on H_2 have diagonal matrices relative to basis $\mathcal{B}_2$ such that $C_1(i,i) \le C_2(i,i)$, for all $i \ge 1$. Then,

$$\mathrm{trace}[GC_1G^*] \le \mathrm{trace}[GC_2G^*] \quad \text{for all } G \text{ in } \mathcal{L}(H_2, H_1).$$

Proof:

$$\begin{aligned} 0 &\le \|G(C_2 - C_1)^{1/2}\|_{HS}^2 \\ &= \mathrm{trace}[G(C_2 - C_1)^{1/2}][G(C_2 - C_1)^{1/2}]^* \\ &= \mathrm{trace}[G(C_2 - C_1)G^*] \\ &= \mathrm{trace}[GC_2G^*] - \mathrm{trace}[GC_1G^*], \end{aligned}$$

since $C_2 - C_1$ is a nonnegative-definite self-adjoint operator. ■

Lemma 4. Suppose K is uncorrelated relative to $\mathcal{B}_1$ and $\mathcal{B}_2$. Then for any $G \in \mathcal{L}(H_2, H_1)$ we have:

$$\mathrm{E}\|GKx\|^2 \le \mathrm{trace}[GCG^*],$$
$$\text{where} \quad C = \begin{cases} r^2 T, & \text{if } x \in \mathrm{S}_r; \\ C_{Kb}, & \text{if } x \in \Pi_b; \end{cases}$$

and where $b = (b_1, b_2, \ldots) \in H_1$ and T is a trace class operator on H_2 whose matrix is diagonal relative to $\mathcal{B}_2$ and is given by: $T(i,i) = \sigma_i^2 = \sup_{j \ge 1} \sigma_{ij}^2$, and C_{Kb} is the covariance operator of Kb.

Proof: From lemma 2 it follows C_{Kx} has diagonal matrix with respect to $\mathcal{B}_2$ and:

$$C_{Kx}(i,i) = \textstyle\sum_n x_n^2 \sigma_{in}^2 \le \sigma_i^2 \sum_n x_n^2 = \sigma_i^2 \|x\|^2. \tag{7}$$

Now, $\|x\| \le r$ implies $C_{Kx}(i,i) \le r^2\sigma_i^2 = r^2 T(i,i)$. Lemma 3 applies to C_{Kx} and r^2T to give:

$$\mathrm{trace}[GC_{Kx}G^*] \le \mathrm{trace}[GCG^*],$$

for all G in $\mathcal{L}(H_2, H_1)$, where $C = r^2 T$. This completes the proof for the constraint $x \in \mathrm{S}_r$.

On the other hand, if $x \in \Pi_b$, from (7) we get: $C_{Kx}(i,i) = \sum_n x_n^2 \sigma_{in}^2 \le \sum_n b_n^2 \sigma_{in}^2 = C_{Kb}(i,i)$. So, $C_{Kx}(i,i) \le C_{Kb}(i,i)$ and once again lemma 3 applies to give: $\mathrm{trace}[GC_{Kx}G^*] \le \mathrm{trace}[GCG^*]$, for all G in $\mathcal{L}(H_2, H_1)$, and this time $C = C_{Kb}$. So in either case we have:

$$\mathrm{E}\|GKx\|^2 = \mathrm{trace}[GC_{Kx}G^*] \le \mathrm{trace}[GCG^*],$$

and C is given according to whether $x \in \mathrm{S}_r$ or $x \in \Pi_b$, as in the lemma statement. ■

The following theorem concludes the matter of upper-bounding the error for both norm and parallelepiped constraints. The main job has been done in lemmas 1,2,3 and particularly 4.

Theorem 2. Suppose K is uncorrelated and x is either in S_r or in Π_b. Then in the estimation problem $y = Fx + w$, $\hat{x} = Gy$, for each G in $\mathcal{U}$, the mean-square error is no greater than

$$\mathrm{trace}[G(C + C_w)G^*],$$

where C depends on how x is constrained and is determined as in lemma 4. The bound is minimized by:

$$G_o = \left(\left[(C + C_w)^{-1/2} A \right]^+ (C + C_w)^{-1/2} \right)^{**},$$

in which it is assumed $(C + C_w)^{-1/2} A$ is densely defined and $\left[(C + C_w)^{-1/2} A\right]^+ (C + C_w)^{-1/2}$ is continuous on the dense domain $\mathrm{ran}\,(C + C_w)^{1/2}$. The superscripts $+$ and $**$ denote, respectively, the pseudoinverse and the continuous extension.

Proof: Lemma 4 implies $\mathrm{E}||GKx||^2 \leq \operatorname{trace}[GCG^*]$ and we always have: $\mathrm{E}||Gw||^2 = \operatorname{trace}[GC_wG^*]$. Therefore, $e = \mathrm{E}||GKx||^2 + \mathrm{E}||Gw||^2 \leq \operatorname{trace}[G(C+C_w)G^*]$. The statement about G_o follows from an application of lemma 1. ∎

Corollary. If H_1 is finite dimensional and $C + C_w$ is onto, or if H_2 is finite dimensional, we have:

$$\begin{aligned} G_o &= [A^*(C+C_w)^{-1}A]^{-1}A^*(C+C_w)^{-1}, \\ \text{minimum value of error bound } &= \operatorname{trace}[A^*(C+C_w)^{-1}A]^{-1}, \\ \text{corresponding error } &= \operatorname{trace}[G_o(C_{Kx}+C_w)G_o^*]. \end{aligned}$$

Proof: Corollary to lemma 1. ∎

When H_2 and H_1 are ∞-dimensional, it is necessary to ensure the upper bound is finite. So we require that for the norm constraint problem, T be trace-class, i.e., $\sum_i \sigma_i^2 < \infty$. Here is a simple example that shows the conditions imposed on K in the norm constraint part of theorem 2 are not inconsistent. We construct an infinite random matrix which in the standard basis of ℓ_2 represents an operator that has all the properties required of K. Consider a sequence of uncorrelated, zero mean, unit variance real-valued random variables. Arrange this sequence into an infinite matrix through some bijection of positive integers onto ordered pairs of positive integers, to obtain a matrix of random variables $\{Z_{ij},\ i,j = 1,2,\dots\}$. Define another matrix K, by $K_{ij} = Z_{ij}/ij$, for $i,j \geq 1$. Then $\sigma_{ij}^2 = \operatorname{Var}(K_{ij}) = 1/(ij)^2$ and $\sigma_i^2 = 1/i^2$. On the other hand, $\sum_{ij} K_{ij}^2 < \infty$, $[\mu]$-a.s., because its expectation is finite. So, $K(\omega)$ is compact, has zero mean and uncorrelated entries, and satisfies $\sum_i \sigma_i^2 < \infty$.

ILL-POSED CASE

Existence of unbiased estimators depends on whether operator A has closed range or not. In general, A has a bounded left inverse, i.e., there is an unbiased estimator, if and only if A is bounded below, i.e., if and only if $\operatorname{ran} A$ is closed and $\ker A$ is zero. In this part we consider a class of problems in which A has zero kernel but non-closed range. For A to have non-closed range it is necessary that it not be finite-rank. Throughout this section H_1 and H_2 are infinite dimensional real separable H-spaces and A is not finite-rank.

The class of problems considered here consists of those in which A is compact. In fact, if a compact operator A had closed range, then $\operatorname{ran} A$ would have to be finite-dimensional, contradicting the assumption that A is not finite-rank; see statements just before theorem 1. It should be mentioned that compactness is only a sufficient condition for the range to be non-closed. But as far as applications is concerned, the compact case does represent a fairly large subclass of ill-posed problems because most applied problems come from an integral operator formulation with L^2 kernels and hence give rise to Hilbert-Schmidt operators which are contained in the set of compact operators.

What has been said so far proves that if A is compact it does not have a continuous left inverse. But here is yet another argument which is shorter and nicer: If A is compact and G is continuous then GA is compact and cannot equal I_{H_1} because I_{H_1} is not compact. So G cannot be unbiased. In fact, not only $\mathrm{I}_{H_1} \neq GA$, we even have $||\mathrm{I}_{H_1} - GA|| \geq 1$. A proof of this statement uses a property of compact operators saying that if GA is compact and $(h_n)_{n=1}^{\infty}$ is any orthonormal sequence in H_1, then $\lim_{n\to\infty} ||GAh_n|| = 0$. This means we cannot have continuous estimators that are almost unbiased in the sense that:

$$\mathcal{U}_\varepsilon \overset{\mathrm{d}}{=} \{G \in \mathcal{L}(H_2, H_1) : ||\mathrm{I}_{H_1} - GA|| \leq \varepsilon\} = \emptyset, \qquad \text{for} \quad 0 \leq \varepsilon < 1.$$

To continue, we discard the unbiasedness condition on G and require only that G be in $\mathcal{L}(H_2, H_1)$. The error formula is the one given before:

$$e = \mathrm{E}||x - \hat{x}||^2 = ||x - GAx||^2 + \mathrm{E}||GKx||^2 + \mathrm{E}||Gw||^2. \tag{1}$$

However, without unbiasedness, a head-on minimization of any upperbound is likely to result in bad estimators. For example, we may write:

$$e \leq (||x|| + ||GAx||)^2 + \mathrm{E}||GKx||^2 + \mathrm{E}||Gw||^2;$$

and $G=0$ minimizes the right hand side, for all x in H_1.

Singular Decomposition and Linear Precision Gauge.

The material that follows is taken almost verbatim from [2] and [6]. The method of Linear Precision Gauge is devised to treat ill-posed problems in a judicious manner. It consists of an increasing sequence of finite dimensional subspaces of the parameter space H_1 which converges to it. There is a certain amount of freedom in choosing the sequence under discussion, however it is proved that the LPG corresponding to the *Singular Decomposition* or *Schmidt Decomposition* of A is, in a sense, most "efficient". The reader is refered to [2] for details. Here we shall use only the LPG given by the singular decomposition of A and refer to it as the *canonical* linear precision gauge.

This is a summary of singular decomposition of A: Both A^*A and $(AA^*)_r$ are compact self-adjoint positive-definite operators; where $(AA^*)_r$ is the restriction to $\overline{\operatorname{ran} A}$ of AA^*. The spectral theory of such operators says there is an orthonormal basis $(\phi_n)_{n=1}^{\infty}$ for $\overline{\operatorname{ran} A^*A} = H_1$, comprised of eigenvectors of A^*A and an orthonormal basis $(\psi_n)_{n=1}^{\infty}$ for $\overline{\operatorname{ran}(AA^*)_r} = \overline{\operatorname{ran} A}$, comprised of eigenvectors of $(AA^*)_r$. Also, $(AA^*)_r$ and A^*A have the same sequence of eigenvalues $(\lambda_1,\lambda_2,\ldots)$ which are positive and converge monotonically decreasing to zero. The eigenvectors ϕ's and ψ's are related through $A\phi_n = \sqrt{\lambda_n}\psi_n$ and $A^*\psi_n = \sqrt{\lambda_n}\phi_n$, for all $n \geq 1$.

For each positive k, we let N_k and M_k be, respectively, the linear spans of $\phi_1,\ldots,\phi_k$ and $\psi_1,\ldots,\psi_k$. It is clear that $M_k = A(N_k)$. Also, define orthogonal projections Q_k and P_k of H_1 and H_2 onto N_k and M_k, respectively. Dimensions of N_k and M_k are the same and equal k. Consider the orthogonal projection of the observed data onto M_k. The important parameters are:

$$y_k = P_k y, \quad w_k = P_k w, \quad x_k = Q_k x \quad \text{and} \quad A_k = A|_{N_k}.$$

The original problem: $y = Fx + w$, $\hat{x} = Gy$, $G \in \mathcal{L}(H_2, H_1)$, is replaced by the kth problem which is given by the pair of equations:

$$\begin{aligned} y_k &= A_k x_k + P_k K x + w_k, \\ \widehat{x_k} &= G_k y_k, \quad G_k \in \mathcal{L}(M_k, N_k). \end{aligned}$$

Define ϵ_k to be the mean-square error for kth estimation problem: $\epsilon_k = \mathrm{E}\|x_k - \widehat{x_k}\|^2$. A routine calculation, using the fact that $P_k A Q_k^{\perp} = 0$, gives:

$$\epsilon_k = \|x_k - G_k A_k x_k\|^2 + \mathrm{E}\|G_k P_k K x\|^2 + \mathrm{E}\|G_k w_k\|^2. \tag{8}$$

We want G_k to be unbiased, i.e., $G_k A_k = \mathrm{I}_{N_k}$. Since $\dim N_k = \dim M_k = k$ and A_k is nonsingular, we must have $G_k = A_k^{-1}$ and from (8) we get:

$$\epsilon_k = \mathrm{E}\|A_k^{-1} P_k K x\|^2 + \mathrm{E}\|A_k^{-1} w_k\|^2 \tag{9a}$$

$$= \sum_{i=1}^{k} \frac{\mathrm{E}|\langle Kx, \psi_i\rangle|^2 + \mathrm{E}|\langle w, \psi_i\rangle|^2}{\lambda_i}. \tag{9b}$$

From (9b) it can be seen the sequence $(\epsilon_k)_{k=1}^{\infty}$ is monotone increasing. The m.s.e. for the actual image, i.e., $\mathrm{E}\|x - \widehat{x_k}\|^2$ can be shown to be $\|Q_k^{\perp} x\|^2 + \epsilon_k$. The resolution error is $\|Q_k^{\perp} x\|^2$ which decreases with k. These were the components of what is known as the canonical linear precision gauge.

Now, we would like to have $\|Q_k^{\perp} x\|$ as small as possible, which means we want k to be as large as possible. But as mentioned before, increasing k also increases the noise error ϵ_k. So the question is how large can k get before ϵ_k becomes intolerable. This question cannot be answered in its present form because ϵ_k depends on the unknown x through K. We now derive an x-free upperbound for ϵ_k, subject to the norm or parallelepiped constraints and for the case when random operator K is uncorrelated. It was seen in lemma 4 that:

$$\mathrm{E}\|GKx\|^2 = \operatorname{trace}[GC_{Kx}G^*] \leq \operatorname{trace}[GCG^*], \quad \text{if } K \text{ is uncorrelated and } G \in \mathcal{L}(H_2, H_1) \tag{10}$$

where C does not depend on x but it depends on whether x is in S_r or in Π_b and in either case C is defined in terms of operators that have diagonal matrices with respect to bases that make K uncorrelated. But, since the trace functional is bases-independent, once (10) is derived it can be used in any bases. It should be borne in mind though that Π_b is aligned with respect to the basis which makes K uncorrelated and not the basis $(\phi_n)_{n=1}^{\infty}$ given by the singular decomposition of A. Equation 9a can be written as:

$$\epsilon_k = \mathrm{trace}[A_k^{-1} P_k C_{Kx} P_k (A_k^*)^{-1}] + \mathrm{trace}[A_k^{-1} P_k C_w P_k (A_k^*)^{-1}],$$

taking G in (10) to be $A_k^{-1} P_k$ we get:

$$\begin{aligned}\epsilon_k &\leq \mathrm{trace}[A_k^{-1} P_k C P_k (A_k^*)^{-1}] + \mathrm{trace}[A_k^{-1} P_k C_w P_k (A_k^*)^{-1}] \\ &= \mathrm{trace}[A_k^{-1} P_k (C + C_w) P_k (A_k^*)^{-1}],\end{aligned}$$

which with some manipulation becomes:

$$= \sum_{i=1}^{k} \frac{\langle (C + C_w)\psi_i, \psi_i \rangle}{\lambda_i}; \tag{11}$$

where C is as in lemma 4. We define $\alpha_k = \sum_{i=1}^{k} \langle (C + C_w)\psi_i, \psi_i \rangle / \lambda_i$, $k \geq 1$, as the sequence of upper bounds sought for ϵ_k, $k \geq 1$. Observe that $(\alpha_k)_{k=1}^{\infty}$ is also monotone increasing.

Suppose there is given an increasing sequence $(t_k)_{k=1}^{\infty}$ of tolerable noise errors or thresholds, for stages 1,2, etc. The idea is to choose the "optimal index" k_o such that $\alpha_k \leq t_k$ for all $k \leq k_o$ but $\alpha_{k_o+1} > t_{k_o+1}$. Let $\alpha = \lim \alpha_k$. Suppose $\alpha = \infty$. Then as k increases, the noise error gets large without bound. The optimal index should be chosen according to:

$$k_o = \inf\{k : \alpha_k > t_k\} - 1, \tag{12}$$

assuming $\alpha_1 \leq t_1$ to be able to start at all, and also assuming there exists a $k \geq 1$ such that $\alpha_k > t_k$; otherwise $k_o = \infty$. Suppose, on the other hand, that $\alpha < \infty$. If the value of α is an acceptable noise level or if $\alpha_k \leq t_k$, $\forall k \geq 1$, again $k_o = \infty$, i.e., we are free to choose k as large as we like; otherwise (12) should be used to arrive at the proper index k_o. Once k_o is chosen, the estimator is: $G_o = A_{k_o}^{-1} P_{k_o}$ and the estimate:

$$\hat{x} = \widehat{x_{k_o}} = G_o(y) = A_{k_o}^{-1} P_{k_o}(y) = A_{k_o}^{-1} \sum_{i=1}^{k_o} \langle y, \psi_i \rangle \psi_i = \sum_{i=1}^{k_o} \langle y, \psi_i \rangle A_{k_o}^{-1} \psi_i,$$

but $A_{k_o}^{-1} \psi_i = \frac{1}{\sqrt{\lambda_i}} \phi_i$, so,

$$\hat{x} = \sum_{i=1}^{k_o} \frac{1}{\sqrt{\lambda_i}} \langle y, \psi_i \rangle \phi_i. \tag{13}$$

Remark. Except for the pseudoinverse estimator, the estimators given in theorems 1 and 2 as well as the one in (13) for the ill-posed case, are constrained estimators, but the estimate $\hat{x}$ need not wind up in the constraint set. It can be shown that the mean-square-error is improved if the suboptimal estimator G_o is followed by a projection of $\hat{x}$ onto the closed convex set Π_b or S_r. With this additional step the final estimator becomes nonlinear but is still continuous.

APPENDIX

Given a probability space $(\Omega, \mathcal{F}, \mu)$ and a B-space X, there are at least three kinds of measurability for mappings of Ω into X: Borel, weak and strong. While a probabilistic notion such as independence has traditionally been formulated in terms of Borel measurability, an analytical concept such as Bochner Integral (integration of X-valued functions on Ω), seems to have been treated entirely in the context of weakly and strongly measurable functions. Since we have occasion to deal with both independence and integration (expectation), among a few other elementary operations, and since Borel, weak and strong measurability are not in general equivalent, we would like to unify these concepts in order to avoid extra or possibly conflicting hypotheses and to ensure proper theoretical framework. This is acheived by assuming X is separable for, then, the three concepts coincide; [8], pp 33 and [3], pp 72-73. In the paper, a random variable is defined as a measurable mapping of Ω into a separable H-space and the word measurable is used to imply any one of the three notions.

A similar problem arises in defining random linear operators. Given two B-spaces X and Y, a random operator T can be defined as a measurable mapping of Ω into $\mathcal{L}(X,Y)$, or, as a mapping for which $T(\cdot)x$ is a Y-valued measurable function on Ω, for all x in X; [3], def. 3.5.5 pp 74. Both definitions have desirable implications which enable one to carry out computations that would be routine for a finite dimensional problem but need justification when the spaces are ∞-dimensional. If X and Y are separable H-spaces, as is the case in the paper, and if we assume $T : \Omega \longrightarrow \mathcal{L}(X,Y)$ has compact realizations, i.e., $\operatorname{ran} T \subset \mathcal{L}_o(X,Y) \subset \mathcal{L}(X,Y)$, where $\mathcal{L}_o(X,Y)$ is the set of compact linear maps from X into Y, then it follows from Dunford's theorem, [3], thm. 3.5.5 pp 74-75, that the two definitions are equivalent. Actually this is a way of making T "almost separably-valued", [3], def. 3.5.3 pp 72. It can be shown if X and Y are separable H-spaces, $\mathcal{L}_o(X,Y)$ is a separable B-space, and $T(\omega)$'s being compact for each ω in Ω is a sufficient condition for T to be almost separably-valued. In this paper we define a random operator as an $\mathcal{L}_o(X,Y)$-valued measurable function on Ω where X and Y are real separable H-spaces.

Theorem 1. Let X and Y be real separable H-spaces, T be a random operator on X into Y and u be an X-valued random variable. Then Tu defined by $T(\omega)u(\omega)$, for all ω in Ω, is a Y-valued random variable. Furthermore, if T and u are independent and integrable, then Tu is also integrable and $\mathrm{E}(Tu) = (\mathrm{E}T)(\mathrm{E}u)$.

Proof: There are sequences $(T_n)_{n=1}^{\infty}$ and $(u_n)_{n=1}^{\infty}$ of countably valued measurable functions which converge almost surely to T and u respectively. For each ω we can write:

$$\|(Tu)(\omega) - (T_n u_n)(\omega)\| \leq \|T(\omega) - T_n(\omega)\| . \|u(\omega)\| + \|T_n(\omega)\| . \|u(\omega) - u_n(\omega)\|,$$

which implies $T_n u_n \longrightarrow Tu$, $[\mu]$-a.s., since $(\|T_n(\omega)\|)_{n=1}^{\infty}$ is a bounded sequence for each ω. This proves Tu is measurable. For the second half, we use the fact that if T and u are integrable, then $(T_n)_{n=1}^{\infty}$ and $(u_n)_{n=1}^{\infty}$ satisfy: $\lim_{n\to\infty} \mathrm{E}\|T - T_n\| = \lim_{n\to\infty} \mathrm{E}\|u - u_n\| = 0$, $\mathrm{E}T \overset{\mathrm{d}}{=} \lim \mathrm{E}T_n$ and $\mathrm{E}u \overset{\mathrm{d}}{=} \lim \mathrm{E}u_n$. Also, (T_n) and (u_n) are constructible from T and u, respectively, and so are independent; [3], proofs of theorems 3.5.3 and 3.7.4 and their corollaries , pp 72-81. Integrating both sides of the above inequality gives:

$$\mathrm{E}\|Tu - T_n u_n\| \leq \mathrm{E}\|T - T_n\| . \mathrm{E}\|u\| + \mathrm{E}\|T_n\| . \mathrm{E}\|u - u_n\|,$$

which reads $\lim_{n\to\infty} \mathrm{E}\|Tu - T_n u_n\| = 0$. Therefore, $\mathrm{E}(Tu) \overset{\mathrm{d}}{=} \lim_{n\to\infty} \mathrm{E}(T_n u_n)$. Using independence of T_n and u_n we can show $\mathrm{E}(T_n u_n) = (\mathrm{E}T_n)(\mathrm{E}u_n)$, for all $n \geq 1$. This now proves that $\mathrm{E}(Tu) = (\mathrm{E}T)(\mathrm{E}u)$. ∎

Theorem 2. Suppose u and v are integrable independent random variables. Then, $\mathrm{E}\langle u,v\rangle = \langle \mathrm{E}u,\mathrm{E}v\rangle$.

Proof: Let $(\phi_n)_{n=1}^{\infty}$ be an orthonormal basis. We always have $\mathrm{E}u = \sum_n \langle \mathrm{E}u,\phi_n\rangle \phi_n$. Hille's theorem implies $\langle \mathrm{E}u,\phi_n\rangle = \mathrm{E}\langle u,\phi_n\rangle = \mathrm{E}u_n$; [3], pp 83. So, $\langle \mathrm{E}u,\mathrm{E}v\rangle = \sum_n (\mathrm{E}u_n)(\mathrm{E}v_n)$. On the other hand, $\langle u,v\rangle = \sum_n u_n v_n$. So, $\mathrm{E}\langle u,v\rangle = \mathrm{E}\sum_n u_n v_n = \sum_n \mathrm{E}(u_n v_n) = \sum_n (\mathrm{E}u_n)(\mathrm{E}v_n)$; using Lebesgue's dominated convergence theorem and independence of u and v. ∎

ACKNOWLEDGMENT

The research reported in this paper was supported by the Office of Naval Research, grant number N00014-85-K-0149.

BIBLIOGRAPHY

1. Beutler, F.J. & Root, W.L., "The Operator Pseudoinverse in Control and System Identification," ed. M.Z. Nashed, *Generalized Inverses and Applications,* Academic Press, New York 1976
2. Joyce, L.S. & Root, W.L., "Precision Bounds in Superresolution Processing," *J.Opt. Soc. Am.,* A/vol. 1, No. 2, pp 149-168, February 1984
3. Hille & Phillips, *Functional Analysis and Semigroups,* AMS, Colloquium Publications, vol. 31, 1957
4. Nashed & Salehi, "Measurability of Generalized Inverses of Random Linear Operators," *SIAM J. Appl. Math.,* vol. 25, No. 4, December 1973
5. Riesz & Nagy, *Functional Analysis,* Frederick Ungar Pub. Co. 1955
6. Root, W.L., "Ill-posedness and Precision in Object-Field Reconstruction Problems," *J. Opt. Soc. Am.,* A/vol. 4, pp 171-179, January 1987
7. Schatten, R., *Norm Ideals of Completely Continuous Operators,* Springer-Verlag, Berlin 1960
8. Scheidt & Purkert, *Random Eigenvalue Problems,* North-Holland, 1983
9. Twerskey, V., "On Propagation in Random Media of Discrete Scatterers," ed. Bellman, R., *Proc. Symp. Appl. Math.,* vol. 16, AMS Providence, RI 1964

SOME COMMENTS ON ESTIMATION OF RANDOM VARIABLES

A. E. WESSEL and G. L. WISE
Department of Electrical and Computer Engineering and Department of Mathematics
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

The martingale convergence theorem often arises in attempts to establish the convergence of a sequence of estimates of a random variable. In this paper some consequences of a careless approach to the use of the martingale convergence theorem in estimating a random variable are presented. It is shown that the martingale convergence theorem might be of no help whatsoever in estimating a zero mean, unit variance Gaussian random variable in terms of a sequence of other Gaussian random variables. Results on avoiding these undesired consequences are then presented.

INTRODUCTION

The martingale convergence theorem is one of the key theorems of probability theory, and it often surfaces in applications. Basic results on martingale theory can be found in [1, Chap. 7], [2, Sections 32 and 39], and [3, Chap. 7]. In the context of estimation theory, the martingale convergence theorem often arises as a useful tool in establishing convergence of many popular estimation schemes. For example, we might be concerned with a sequence of random variables which represent our attempts to estimate a quantity of interest as more data is processed. In this situation we might encounter random variables such as $E(X|Y_1, Y_2, \ldots, Y_k)$, where X is a second order random variable we are attempting to estimate from the first k random variables of a sequence $\{Y_n\}$ of random variables. In this situation, the martingale convergence theorem establishes the mean square as well as the almost sure convergence of the sequence of second order random variables $\{E(X|Y_1, Y_2, \ldots, Y_k)\}$.

In this paper we present some examples showing that the convergence guaranteed by the martingale convergence theorem can be utterly useless. These examples illustrate some of the many pitfalls that await an unwary investigator in estimation theory. We then close with some comments on avoiding the situation illustrated by the examples.

DEVELOPMENT

Let **R** denote the real numbers, and let **N** denote the natural numbers, or positive integers. Let $\underline{\mathbf{P}}(\mathbf{R})$ denote the power set of **R**, and let Σ denote the Borel subsets of **R**.

For any measure M on $(\mathbf{R}, \Sigma)$, let M_* and M^* denote, respectively, the inner and outer M measures on $(\mathbf{R}, \underline{\mathbf{P}}(\mathbf{R}))$ given by

$$M_*(A) = \sup \{M(B): B \in \Sigma \text{ and } A \supset B\} \text{ and}$$
$$M^*(A) = \inf \{M(B): B \in \Sigma \text{ and } B \supset A\}, \text{ for } A \in \underline{\mathbf{P}}(\mathbf{R}).$$

The following equality can easily be established (see, for example, [4, p. 61]). For $B \in \Sigma$ and $A \in \underline{\mathbf{P}}(\mathbf{R})$

$$M_*(A \cap B) + M^*(A^c \cap B) = M(B). \tag{1}$$

For a set $A \in \underline{\mathbf{P}}(\mathbf{R})$, let I_A denote the indicator function of A, and let $-A = \{x \in \mathbf{R}: -x \in A\}$. For any two subsets A and B of $\mathbf{R}$, let A\B denote those elements in A which are not in B, and let $A \Delta B$ denote $(A \backslash B) \cup (B \backslash A)$. Recall that two measures on a given measurable space are said to be equivalent if they assign zero measure to precisely the same measurable sets. Let P denote a probability measure on $(\mathbf{R}, \Sigma)$ such that $P(B) = P(-B)$ for all $B \in \Sigma$ and such that P is equivalent to Lebesgue measure on $(\mathbf{R}, \Sigma)$. Thus $(\mathbf{R}, \Sigma, P)$ is a probability space. We note that such measures P exist in abundance; for example, any symmetric probability density function that is everywhere positive gives rise to such a measure P. Furthermore, Gaussian measure on $(\mathbf{R}, \Sigma)$ as presented in Example 3 satisfies these conditions.

Lemma 1: There exists a subset S of $\mathbf{R}$ such that

$$P_*(S) = 0 = P_*(S^c). \tag{2}$$

Proof: Let λ denote Lebesgue on $(\mathbf{R}, \Sigma)$. Let $S \in \underline{\mathbf{P}}(\mathbf{R})$ be the the set constructed in [4, p. 70]; that is, $\lambda_*(S \cap B) = 0$ and $\lambda^*(S \cap B) = \lambda(B)$ for $B \in \Sigma$. Recall that on the measurable space $(\mathbf{R}, \Sigma)$, P is equivalent to Lebesgue measure λ; that is, $P(B) = 0$ if and only if $\lambda(B) = 0$. It immediately follows that $P_*(S) = 0$.

We first claim that $P_*(S^c \cap J) = 0$ for all bounded subintervals J of $\mathbf{R}$. Fix such an interval J. We know that $\lambda^*(S \cap J) = \lambda(J)$; it follows from (1) that $\lambda_*(S^c \cap J) = 0$. Since λ and P are equivalent measures, this first claim is established. It now easily follows that $P_*(S^c) = 0$. Indeed, if $P_*(S^c) > 0$, then $\exists\, B \in \Sigma$ with $S^c \supset B$ and $P(B) > 0$, and we can then find a bounded interval J so that $P(B \cap J) > 0$ and thus $P_*(S^c \cap J) > 0$, contradicting the preceding result.

QED

Lemma 2: Let S be a subset of **R** which satisfies (2). If B_1, B_2, D_1, and D_2 are Borel sets and $(B_1 \cap S) \cup (B_2 \cap S^c) = (D_1 \cap S) \cup (D_2 \cap S^c)$, then $P(B_1 \Delta D_1) = 0$ and $P(B_2 \Delta D_2) = 0$.

Proof: Equation (1) and the choice of S imply that

$$P^*(B \cap S) = P(B) = P^*(B \cap S^c) \tag{3}$$

for all $B \in \Sigma$. By assumption, $(B_1 \cap S) \cup (B_2 \cap S^c) = (D_1 \cap S) \cup (D_2 \cap S^c)$. Intersecting with S, we obtain $(B_1 \cap S) = (D_1 \cap S)$, which implies that $(B_1 \Delta D_1) \cap S = \emptyset$. Since $B_1 \Delta D_1$ is a Borel set, (3) implies that $P(B_1 \Delta D_1) = P^*((B_1 \Delta D_1) \cap S) = P^*(\emptyset) = 0$. Similarly, $P(B_2 \Delta D_2) = 0$.

QED

In Lemma 1 we established the existence of a set S satisfying (2). Now let

$$\mathfrak{S} = \{(S \cap B_1) \cup ((S^c) \cap B_2) : B_1 \text{ and } B_2 \text{ belong to } \Sigma\}.$$

It follows straightforwardly that $\mathfrak{S}$ is a σ-algebra on **R**. It follows from Lemma 2 that we can define a probability measure μ on the measurable space $(\mathbf{R}, \mathfrak{S})$ by

$$\mu((S \cap B_1) \cup ((S^c) \cap B_2)) = [P(B_1) + P(B_2)]/2.$$

Note that μ agrees with P on the Borel sets. Henceforth, we work with the probability space $(\mathbf{R}, \mathfrak{S}, \mu)$.

We first note that on this probability space, for any Borel set B, $\mu(S \cap B) = \mu(S)\mu(B) = (0.5)\mu(B) = \mu(S^c)\mu(B) = \mu((S^c) \cap B)$, and thus S as well as S^c are independent of Σ. We use this fact in the following example which is a slight modification of the construction used in [5].

Example 1: Let the random variable X_1 be defined on the probability space $(\mathbf{R}, \mathfrak{S}, \mu)$ and be given by

$$X_1(x) = I_S(x) - I_{S^c}(x).$$

The random variable X_1 is nowhere equal to zero, $\|X_1\|_{L_1} = \|X_2\|_{L_2} = 1$, yet we claim that $E(X_1|\Sigma)$, the conditional expectation of X_1 with respect to the Borel sets, equals zero. This follows from the fact that X_1 is independent of the Borel measurable sets.

Indeed, $\sigma(X_1)$, the σ-algebra generated by X_1, is $\{S, S^c, \mathbf{R}, \varnothing\}$; and it follows from the comment above that $\sigma(X_1)$ is independent of Σ.

Example 2: Define a random variable X_2 on the probability space $(\mathbf{R}, \mathfrak{I}, \mu)$ by

$$X_2(x) = xI_S(x) - xI_{S^c}(x) . \tag{4}$$

Let $B \in \Sigma$ and notice that $\mu(X_2 \in B) = \mu((S \cap B) \cup ((S^c) \cap (-B)) = [P(B) + P(-B)]/2 = P(B)$. We now observe that $E(X_2|\Sigma) = 0$ a.s. The argument in this case is more involved since $\sigma(X_2)$ is no longer independent of Σ. Indeed, for any Borel set B so that $B = -B$, we have $X_2^{-1}(B) = B$. (In Lemma 4 below, we show that these symmetric Borel sets are essentially the only Borel sets which can be obtained as the inverse image under X_2 of an arbitrary Borel set.) Since symmetric Borel sets of μ (and thus P) measure strictly between zero and one can be found, $\sigma(X_2)$ and Σ cannot be independent since this would require the square of such a number to be equal to itself.

We now show that $E(X_2|\Sigma) = 0$ a.s. For an arbitrary Borel set B, we have that

$$\int_B X_2 \, d\mu = \int_B (xI_S - xI_{S^c}) \, d\mu = \int_{B \cap S} x \, d\mu - \int_{B \cap S^c} x \, d\mu = (1/2) \left[\int_B x \, dP - \int_B x \, dP \right] = 0,$$

where the third equality follows from a straightforward argument based on measurable simple functions and the definition of μ. Now, recall that $E(X_2|\Sigma)$ is that almost surely unique Σ-measurable function $f:\mathbf{R} \to \mathbf{R}$ such that $\int_B X_2 \, d\mu = \int_B f \, d\mu$ for all Borel sets B.

Since $f=0$ satisfies this equation, we see that $f = 0$ a.s.

Example 3: Here we consider a special case of Example 2. For this example, let the probability measure P be given on $(\mathbf{R}, \Sigma)$ by

$$P(B) = (2\pi)^{-1/2} \int_B \exp(-x^2/2) \, dx.$$

Notice that in this case, the identity map on **R** is a Gaussian random variable with zero mean and unit variance. As in Example 2, let X_3 be given by

$$X_3(x) = xI_S(x) - xI_{S^c}(x) .$$

Since, as in Example 2, $\mu(X_3 \in B) = P(B)$ for any Borel set B, we see that X_3 is a Gaussian random variable with zero mean and unit variance.

We now show that any of these three examples demonstrates the potential futility of a careless estimation scheme.

Proposition 3: Let X be a second order random variable defined on $(\mathbf{R}, \Im, \mu)$ satisfying $E(X|\Sigma) = 0$. Then for any sequence $\{Y_k: k \in \mathbf{N}\}$ of Borel measurable functions mapping $\mathbf{R}$ into $\mathbf{R}$ we have that $E(X|Y_1, Y_2, \ldots, Y_k) = 0$ for all $k \in \mathbf{N}$.

Proof: Pick any sequence $\{Y_k: k \in \mathbf{N}\}$ of Borel measurable functions mapping $\mathbf{R}$ into $\mathbf{R}$. Notice that $\{Y_k: k \in \mathbf{N}\}$ is a sequence of random variables on the probability space $(\mathbf{R}, \Im, \mu)$. Consider the martingale $\{X_k = E(X|Y_1, Y_2, \ldots, Y_k): k \in \mathbf{N}\}$, and notice that $\Sigma \supset \sigma(Y_1, Y_2, \ldots, Y_k)$ for all $k \in \mathbf{N}$. Therefore, for any $k \in \mathbf{N}$,

$$X_k = E(X|Y_1, Y_2, \ldots, Y_k) = E(\, E(X|\Sigma) \mid Y_1, Y_2, \ldots, Y_k) = 0 \text{ a.s.}$$

QED

We note that each of the random variables constructed in the above three examples satisfies the hypotheses of this proposition. Hence, we see that the convergence guaranteed by the martingale convergence theorem is useless in estimating any of these random variables in terms of the random variables $\{Y_k: k \in \mathbf{N}\}$. More generally, any estimation scheme based on conditional expectation is doomed to failure.

Notice that if Y_k is a Borel measurable bijection for each $k \in \mathbf{N}$, then X_1, X_2, and X_3 can be expressed precisely by a function of Y_k for any $k \in \mathbf{N}$. This is illustrated in the following example.

Example 4: Consider the probability space $(\mathbf{R}, \Im, \mu)$. Let $\{s_k: k \in \mathbf{N}\}$ be a sequence of positive numbers and let $Y_k(x) = s_k x$. Then for each $k \in \mathbf{N}$, Y_k is a Gaussian random variable with zero mean and variance $(s_k)^2$, and X_3 is a Gaussian random variable with zero mean and unit variance. Note that $E(X_3|Y_1, Y_2, \ldots, Y_k) = 0$ a.s. for all $k \in \mathbf{N}$, and yet for any $k \in \mathbf{N}$,

$$X_3(x) = (Y_k(x)/s_k)[I_S(Y_k(x)/s_k) - I_{S^c}(Y_k(x)/s_k)]$$

for each $x \in \mathbf{R}$.

COMMENTS

A typical estimation scheme might involve estimating a second order random variable X by means of $E(\, X|\Sigma_n)$ where Σ_n is a finite σ-algebra (generated perhaps by the first n terms of a sequence of simple random variables) and hence generated by

$\{A_1(n), \ldots, A_{k_n}(n)\}$, a partition of the underlying probability space, which in this case is **R**. One might expect $E(X|\Sigma_n)$ to converge to X so long as $\max\{\mu(A_1(n)), \ldots, \mu(A_{k_n}(n))\}$ goes to zero as $n \to \infty$ and the partitions $\{A_1(n), \ldots, A_{k_n}(n)\}$ are chosen to consist of suitably "nice" sets such as intervals. Such a scheme can, in fact, be chosen so that the Σ_n's generate the Borel sets; the martingale convergence theorem then guarantees the L_2 as well as the almost sure convergence of $E(X|\Sigma_n)$ to X *so long as* X is known to be Borel measurable. If this is not the case, as the examples presented above indicate, such an estimation scheme can fail miserably.

In each of the examples presented above, note that the constructed random variables are not Borel measurable. Indeed, the only Borel sets generated by X_1 are the trivial ones, $\emptyset$ and **R**, and the only Borel sets generated by X_2 (and hence X_3) are those which are "essentially" symmetric, as shown by the following result.

Lemma 4: Let X_2 be the random variable defined on the probability space $(\mathbf{R}, \mathfrak{I}, \mu)$ by (4). Then the only Borel sets D obtained as the inverse image under X of a Borel set are those which satisfy $P(D\Delta(-D)) = 0$.

Proof: Let $D = X_2^{-1}(B)$, where D and B are Borel sets. By the definition of X_2, we have that $X_2^{-1}(B) = (S\cap B)\cup((S^c)\cap(-B))$. Also, D can be written as $D = (S\cap D)\cup((S^c)\cap D)$. Lemma 2 implies that $P(B\Delta D) = 0$ and $P((-B)\Delta D) = 0 = P(B\Delta(-D))$. It follows that $P(D\Delta(-D)) = 0$.

QED

If we wish to successfully estimate a random variable X by means of conditional expectation of X with respect to a sequence of random variables $\{Y_k: k \in N\}$, these random variables *must* generate $\sigma(X)$. This is not the case in the examples presented above, since the data generate a subset of the Borel sets. This indicates the importance of a careful examination of the underlying probability space associated with a given random variable, instead of merely considering properties derivable from its distribution function. In the examples presented above, it is precisely the σ-algebras generated by the data that determine whether or not the estimation scheme will perform satisfactorily. As shown, even in the ubiquitous Gaussian case, disappointing results can occur. The distribution functions or indeed even the range space of the data are of little concern. Our results immediately extend to the more general case of random vectors or even random objects, that is, measurable functions taking values in an arbitrary measurable space.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research under Grant AFOSR-86-0026.

REFERENCES

[1] J. L. Doob, *Stochastic Processes*. New York: Wiley, 1953.

[2] M. Loeve, *Probability Theory II*, 4th edition. New York: Springer-Verlag, 1978.

[3] R. B. Ash, *Real Analysis and Probability*. New York: Academic Press, 1972.

[4] P. R. Halmos, *Measure Theory*. New York: Van Nostrand, 1950.

[5] G.L. Wise, "A Note on a Common Misconception in Estimation," *Systems and Control Letters*, Vol. 5, pp. 355-356, 1985.

AN ERROR ANALYSIS FOR THE KALMAN FILTER APPLIED TO DATA CORRUPTTED BY MULTIPLICATIVE NOISE

BEN S. CHOW, WILLIAM P. BIRKEMEIER
Department of Electrical and Comptuer Engineering
University of Wisconsin- Madison
Madison Wisconsin 53706

ABSTRACT

The error of applying Kalman filter to the data corrupted by multiplicative noise is recursively computed in two methods. The first method allows comparing this error with the ideal Kalman filter error (when multiplicative noise does not exist), and presents the error deterioration percentage of the multiplicative noise to understand the effect of multiplicative noise in applying the Kalman filter. The second method allows comparing that error with the error of our estimator, and presents the error improvement percentage of our estimator over the Kalman filter when multiplicative noise exists. With the above two analytical forms developed and numerical results computed , furthermore a geometric approach and pictorial interpretation for a simple case are given to understand the role of multiplicative noise in linear estimation.

INTRODUCTION

In the application of the theory of optimal estimation to practical problems in filtering, prediction, and other related areas, the optimality of the estimation process depends upon the completeness of the mathematical model representing dynamics of the system involved. In reality, the modeling errors are usually unavoidable and the optimality is not achieved. For example, if there is a multiplicative noise involved in the measurement model, the results from applying the Kalman filter [1,2], based on two models of the systems: a state variable signal model and a measurement model, which includes additive noise only, are not proper.

The effects of modeling errors of the Kalman filter have been studied in one category [3,4,5]. They mainly studied the effects of a faulty priori statistics of the covariance matrices of the signal generating process and the additive measurement noise. In this paper, the modeling error is assumed to be from a multiplicative noise in the measurement model. Multiplicative noise is important in many measurement cases because the measuring process is usually determined by the coupling between the measuring devices and the measured objects. The most important coupling relation is the multiplicative relation. Specifically, multiplicative noises exist in the following cases: fading or reflection of the transmitted signal over ionospheric channel, and also certain situations involving sampling, gating, or amplitude modulation [7].

In this paper, we first compare the ideal Kalman filter error (for no multiplicative noise) with the real Kalman error (for having multiplicative noise). The former error is well known as error matrices in Kalman filter. The latter error is for the case of applying a regular Kalman filter to the data corrupted by the multiplicative noise. We then define the error deterioration percentage of the multiplicative noise to understand the effect of multiplicative noise on the application of the Kalman filter.

In our previous papers [8,9,10], an optimal linear MMSE estimator has been developed for problems with multiplicative noise involved. In this paper, we then compare the error of our estimator with the real Kalman filter error for the situation where multiplicative noise exists. We then define the error improvement percentage to understand the efficiency of our estimator over the Kalman filter.

Although the analytical forms for the above two cases are successfully achieved (fortunately, recursive forms) and computer simulation are computed, the role of multiplicative noise are hidden behind the equations and numbers. To have a clear picture for the role of the multiplicative noise in the linear estimation, we also give an geometric approach and interpretation for a simple case. The results are both illustrated in a geometric diagram and simple analytical forms.

PROBLEM FORMULATION:

Systems Formulation:

Signal Model:

$$X(k+1) = A_k X(k) + B_k U(k) \tag{1}$$

Multiplicative noise model:

$$r(k+1) = c_k r(k) + d_k v(k) \tag{2}$$

Data Model:

Real Data Model

$$Z(k) = r(k) H_k X(k) + N(k) \tag{3}$$

Simplified Data Model
(Kalman Data Model)

$$Z'(k) = E[r(k)] H_k X(k) + N(k) \tag{4}$$

where 1) $X(k)$, $r(k)$, $N(k)$, and $Z(k)$ are signal, multiplicative noise, additive noise and data respectively.

2) $U(k)$ and $v(k)$ are the generating random sequences for the signal and the multiplicative noise.

3) $X(k)$ is a $n*1$ vector, $r(k)$ is a scalar, and $Z(k)$ is a $m*1$ vector, with the other terms dimensions consistent.

Assumptions:

(A1) The processes U, v, and N, and the initial states $X(0)$ and $r(0)$ are mutually independent. also U, v, and N are all white.

(A2) $x(0)$, $r(0)$, U, v, and N are all with known means and variances.

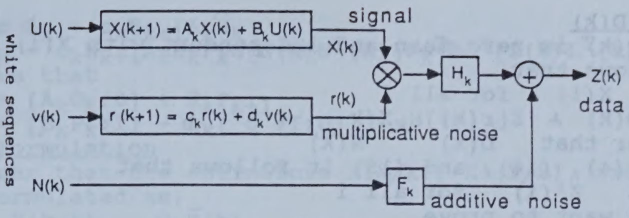


Fig. 1

The notation obeys the following two rules in this paper :

1) Random variables are distinguished from deterministic constants by their time arguments being parenthesized instead of being subscripted.
2) Matrices and vectors are distinguished from scalars by being written in upper cases.

Estimator Choices:

1. $X(k/k)$ is the linear MMSE estimate of $X(k)$ by using our estimator [10] on the data $Z(1), Z(2), ..., Z(k)$.
2. $X''(k/k)$ is the estimate of $X(k)$ by using the Kalman filter incorectly on the data $Z(1), Z(2), ..., Z(k)$.
3. $X'(k/k)$ is the estimate of $X(k)$ by using the Kalman filter correctly on the data $Z'(1), Z'(2), ..., Z'(k)$

Computation Goals:

1. Compute the error matrices P_k for $X(k/k)$ and P''_k for $X''(k/k)$ and then compute the mean square errors e_k and e_k'' for $X(k/k)$ and $X''(k/k)$.

i.e. $P_k = E[\,(X(k/k)-X(k))\,(X(k/k)-X(k))^T\,]$ (5)

$P''_k = E[\,(X''(k/k)-X(k))\,(X''(k/k)-X(k))^T\,]$ (6)

$e_k = \mathrm{Trace}(P_k)$ (7)

$e''_k = \mathrm{Trace}(P''_k)$ (8)

Note P'_k is the error matrix for the Kalman Filter and is well known [2].

2. Compute $\%'_k$ and $\%_k$.

$\%'_k$ is the error deterioration percentage of multiplicative noise compared to no multiplicative noise when the the Kalman Filter is applied.

$\%_k$ is the error improvement percentage of our estimator over the Kalman Filter when Multiplicative noise is involved.

i.e. $\%_{k'} = \mathrm{Trace}(P''_k)/\mathrm{Trace}(P'_k) - 1$ (9)

$\%_k = \mathrm{Trace}(P''_k)/\mathrm{Trace}(P_k) - 1$ (10)

PRELIMINARY

Relation Between Real Data and Simplified Data

Decompose the multiplicative noise $r(k)$ into the mean term $E[r(k)]$ and a zero-mean error term $r'(k)$.

$r(k) = E[r(k)] + r'(k)$. (11)

Substituting (11) into (3), it follows

$Z(k) = Z'(k) + D(k)$ (12)

where $D(k) = H_k r'(k) X(k)$ (13)

Property of D(k)

Because r'(k) is zero mean and independent with X(i) for all i, it follows that

$$D(k) \perp X(i) \quad \text{for all } i$$

Hence, $$D(k) \perp E[r(k)]H_kX(k \tag{14}$$

It is clear that $$D(k) \quad N(k) \tag{15}$$

Combining (4), (14), and (15) it follows that

$$D(k) \perp Z'(i) \quad \text{for all } i \tag{16}$$

We next want to prove

$$D(k+1) = c_kH_{k+1}A_kH_kD(k) + U'(k) \tag{17}$$

$$\begin{aligned} \text{where } U'(k) = {} & d_kv(k)H_{k+1}A_kX(k) + c_kr'(k)H_{k+1}B_kU(k) \\ & + d_kv(k)H_{k+1}B_kU(k) \end{aligned} \tag{18}$$

This is because

$$E[r(k+1)] = c_kE[r(k)] \tag{19}$$

Substracting (19) from (2), it follows that

$$r'(k+1) = c_kr'(k) + d_kv(k) \tag{20}$$

It is clear that

$$H_{k+1}r'(k+1) = H_{k+1}c_kr'(k) + H_{k+1}d_kv(k) \tag{21}$$

Multiplying (1) by (21), it follows that

$$H_{k+1}r'(k+1)X(k+1) = c_kH_{k+1}A_kr'(k)X(k) + U'(k) \tag{22}$$

It is clear that

$$D(k+1) = c_kH_{k+1}A_kr'(k)X(k) + U'(k)$$

Because $\mathrm{Null}(H_k) \subset \mathrm{Null}(H_{k+1}A_k)$, it follows that

$$D(k+1) = c_kH_{k+1}A_k\hat{H}_kD(k) + U'(k),$$

which is Eq. (17).

Note that

$$U'(i) \perp D(k) \quad \text{for all } i < (k) \tag{23}$$

The relationship in (23) can be proved by multiplying (18) by (13), and then taking expectation.

Recursive Matrix Representation of Estimator

Express X(k/k) in terms of the linear combination of Z(i)'s:

$$\begin{aligned} X(k/k) &= \sum g_{k,i}Z(i) \\ &= [g_{k,1}, \ldots, g_{k,k}]\ [Z(1)^T \ldots, Z(k)^T]T \\ &= G_kZ(k) \end{aligned} \tag{24}$$

$$\text{where } G_k = [g_{k,1}, \ldots, g_{k,k}] \tag{25}$$

$$Z(k) = [Z(1)^T, \ldots, Z(k)^T]^T \tag{26}$$

Let L(1), L(2) ,..., and L(k) be the innovation process of Z(2), ..., and Z(k). Similarly L'(k) is the innovation of Z(k) Note that L(k) and L'(k) can be expressed as the linear combination of Z(1), ..., and Z(k).

$$L(k) = P_k\vec{Z}(k) \tag{27}$$

$$L'(k) = P'_k\vec{Z}'(k) \tag{28}$$

In the Kalman Filter, it is true that

$$X'(k+1/k+1) = A_kX'(k/k) + K'_kL'(k+1) \tag{29}$$

where K'_k is the Kalman Gain

$$L'(k+1) = Z'(k+1) - E[r(k+1)]H_{k+1}A_kX(k/k) \tag{30}$$

Combining (24), (28), (29), and (30), the recursive matrix presentation follows:

$$G'_{k+1} = [A_kG'_k : 0] + K'_kP'_{k+1} \tag{31}$$

$$P'_{k+1} = [-E[r(k+1)]H_{k+1}A_kG'_k : I] \tag{32}$$

Similarly, for our estimator

$$X(k+1/k+1) = A_kX(k/k) + K_kL(k+1) \tag{33}$$

$$\text{where } K_k = E[X(k+1)L^T(k+1)]E^{-1}[L(k+1)L^T(k+1)] \tag{34}$$

$$L(k+1) = Z(k+1) - \alpha_kZ(k) + \beta_kL(k) \tag{35}$$

where $\alpha_k = c_k H_{k+1} A_k \widetilde{H}_k$

$\beta_k = c_k H_{k+1} A_k \widetilde{H}_k F_k E[N(k)N^T(k)] F_k{}^T E^{-1}[L(k)L^T(k)]$

It follows that

$$G_{k+1} = [A_k G_k : 0] + K_k P_{k+1} \tag{36}$$

$$P_{k+1} = [\beta_k P_k : I] - \alpha_k [\ 0\ : I : 0] \tag{37}$$

Estimator Formulation

It is clear that the estimators X(k/k), X'(k/k), and X"(k/k) can be formulated as:

1. $X(k/k) = G_k \vec{Z}(k)$
2. $X'(k/k) = G'_k \vec{Z}'(k)$
3. $X''(k/k) = G''_k \vec{Z}(k)$ is the key formulation, which means applying the regular Kalman filter G'_k to the real data Z(k).

Estimator Decomposition

According to $X''(k/k) = G'_k \vec{Z}(k)$, we can decompose the estimator X"(k/k) by decomposing Z(k) or G'_k as the following:

$$\vec{Z}(k) = \vec{Z}'(k) + \vec{D}(k) \tag{38}$$

$$G'_k = G_k - @G_k \tag{39}$$

$$\text{where } D(k) = [D(1), \ldots, D(k)] \tag{40}$$

$$@G_k = G_k - G'_k \tag{41}$$

It is clear that

$$\begin{aligned} X''(k/k) &= G'_k \vec{Z}'(k) + G'_k \vec{D}(k) \\ &= X'(k/k) + G'_k \vec{D}(k) \end{aligned} \tag{42}$$

According to (42), X"(k/k) can be interpreted as the summation of the output from using the Kalman Filter correctly and the output of the Kalman Filter applied on the noise off-set.

$$\begin{aligned} \text{Also, } X''(k/k) &= G_k \vec{Z}(k) - @G'_k \vec{Z}(k) \\ &= X(k/k) - @G_k \vec{Z}(k) \end{aligned} \tag{43}$$

Interpretation of (43) is that X''(k/k) can be decomposed into the term of our estimator and the term of the error inprovement, the improvement of error by using our estimator over the Kalman filter on the multiplicative noise data.

DERIVATION OF FUNDAMENTAL EQUATIONS

Recursive Computation of P_{k+1}

Because $(X(k+1) - X(k+1/k+1)) \perp X(k+1/k+1)$, it is clear that

$$P_{k+1} = E[\ (X(k+1) - X(k+1/k+1))\ X^T(k+1)\ ] \tag{44}$$

Recall

$$\begin{aligned} X(k+1/k+1) = {} & A_k X(k) + \\ & E[X(k+1)L^T(k+1)]E^{-1}[L(k+1)L^T(k+1)]L(k+1) \end{aligned} \tag{45}$$

Substitute the term X(k+1) in (1), and the term X(k+1/k+1) in (45) into (44). After arranging terms, it follows that

$$\begin{aligned} P_{k+1} = {} & A_k P_k A_k{}^T + B_k E[U(k)U^T(k)] B_k{}^T + \\ & E[X(k+1)L^T(k+1)]E^{-1}[L(k+1)L^T(k+1)]E[X(k+1)L^T(k+1)]^T \end{aligned} \tag{46}$$

The recursive computation of $E[X(k+1)L^T(k+1)]$ and $E[L(k+1)L^T(k+1)]$ are listed in [8].

Recursive Computation of P''_{k+1}

Method I:

We shall compute P''_{k+1} based upon Eq.(42).

According (42), it follows that

$P''_k = E[((X(k)-X'(k/k))-G'_k D(k))\ ((X(k)-X'(k/k))-G'_k D(k))]$ (47)

Because of (14) and (15), it follows that

$D(k) \perp (X(k) - X'(k/k))$ (48)

Because of (48), Eq. (47) leads to the following equation:

$$P''_k = E[(X(k)-X'(k/k))(X(k)-X'(k/k))^T] + E[G'_k \vec{D}(k)\vec{D}^T(k)G'_k] = P'_k + M_k \quad (49)$$

where P'_k is the well-known Kalman error matrix.

$$P'_k = (I - K'_k H_k)(A_k P'_k A_k + B_k E[U(k)U^T(k)]B_k^T)(I-K'_k H_k)^T + K'_k F_k E[N(k)N^T(k)]F^T_k K'^T_k \quad (50)$$

K'_k is the Kalman Gain [1].

$$M_k = E[G'(k)\vec{D}(k)\vec{D}^T(k)G'(k)] \quad (51)$$

The error deterioration percentage $\%'_k$ can be computed as

$$\%'_k = [\mathrm{Trace}(P'_k + M_k) / \mathrm{Trace}(P'_k)] - 1 = \mathrm{Trace}(M_k)/\mathrm{Trace}(P'_k) \quad (52)$$

M_{k+1} will be recursively computed as follows:

Let $D_{\vec{k+1}} = E[D(\vec{k+1})D^T(\vec{k+1})]$ (53)

It is clear that

$$M_{k+1} = G'_{k+1} D_{k+1} G'^T_{k+1} \quad (54)$$

Because of (29) and (30), G'_{k+1} can be recursively computed as

$$G'_{k+1} = [T_k G'_k : K'_k] \quad (55)$$

where $T_k = (I - K'_k E[r(k)]H_{k+1})A_k G'_k$

Write D_{k+1} as the following form,

$$D_{\vec{k+1}} = \begin{bmatrix} D_{\vec{k}} & D^T_{\vec{k},k+1} \\ D_{\vec{k},K+1} & D_{k+1,k+1} \end{bmatrix} \quad (56)$$

where $D_{\vec{k},k+1} = E[D(k+1)\vec{D}(k)^T]$

$D_{k+1,k+1} = E[D(k+1)D(k+1)]$

Combinining (53), (54) and (55), the recursive form of M_{k+1} is

$$M_{k+1} = T_k M_k T_k^T + K'_k D_{k+1,k+1} K'^T_k + R_{k+1} \quad (57)$$

$$R_{k+1} = K'_k D_{\vec{k},K+1} T_k G'_k + (K'_k D_{\vec{k},K+1} T_k G'_k)^T \quad (58)$$

To compute M_{k+1}, we need to compute $D_{k+1,k+1}$ and R_{k+1}.

$D_{k+1,k+1}$ can be recursively computed as below:

Using (17), it follws that

$$D_{k+1,k+1} = S_k D_{k,k} S_k^T + E[U'(k)U'(k)^T] \quad (59)$$

where $S_k = c_k H_{k+1} A_k \tilde{\hat{H}}_k$

R_{k+1} can be computed by $D_{k,k+1}$.

As $D_{k+1,k+1}$ in (59), the recursive form of $D_{k,k+1}$ can be obtained by (17) as below:

$$D_{\vec{k},k+1} = c_k A^k [D_{\vec{k-1},k} : D_{k,k}] \quad (60)$$

Eq.(57), Eq.(58), Eq.(59), and Eq.(60) give the the recursive computation of M_{k+1}.

Method II

According to (43), it follows that

$$P''_k = E[(X(k)-X(k/k))-@G_k\vec{Z}(k))\ ((X(k)-X(k/k))-@G_k\vec{Z}(k))] \quad (61)$$

Since $(X(k)-X(k/k)) \perp \vec{Z}(k)$, it is clear that

$$\begin{aligned} P''_k &= E[(X(k)-X(k/k))(X(k)-X(k/k))^T] + \\ &\quad E[@G_k\vec{Z}(k)\vec{Z}(k)^T @G_k^T] \\ &= P_k + Q_k \end{aligned} \qquad (62)$$

$$\text{where } Q_k = @G_k E[\vec{Z}(k)\vec{Z}^T(k)]@G_k^T \qquad (63)$$

The error improvement percentage $\%_k$ can be computed in terms of P_k and Q_k as below:

$$\begin{aligned} \%_k &= \text{Trace}(P''_k)/\text{Trace}(P_k) - 1 \\ &= \text{Trace}(Q_k)/\text{Trace}(P_k) \end{aligned} \qquad (64)$$

P_k can be recursive computed as (46).

Q_k can be recursively computed as follows.

Combining (55), (41), $@G_{k+1}$ can be recursively computed as

$$@G_{k+1} = [\ A_k @G_k : 0\ ] + (K_k P_{k+1} - K'_k P'_{k+1}) \qquad (65)$$

Note that P_k and P'_k are recursively computed as in (46) and (50) $E[Z(k+1)Z(k+1)^T]$ can be recursively computed as:

$$E[\vec{Z}(k+1)\vec{Z}(k+1)^T] = \begin{bmatrix} E[\vec{Z}(k)\vec{Z}(k)^T] & E[\vec{Z}(k)Z(k+1)^T] \\ E[Z(k+1)\vec{Z}(k)^T] & E[Z(K+1)Z(k+1)^T] \end{bmatrix} \qquad (66)$$

Eq.(65) and Eq.(66) give the recursive computation of Q_{k+1} according to (63).

NUMERICAL EXAMPLE

To illustrate the use of the derived equations, a simple example is simulated here so that the problem can be studied parametrically, and some essential features can be deduced from the results.

$$\begin{aligned} x(k+1) &= 0.9x(k) + u(k) \\ r(k+1) &= r(k) + v(k) \\ z(k) &= r(k)x(k) + n(k) \end{aligned}$$

The random generating sequences u, v, and n are all zero-mean with variances 0.5, 0.1, and 1.0 respectively. The initial states of signal and multiplicative noise x(0) and r(0) are with means both 1.0 , and variances 1.0, 0.1 respectively.

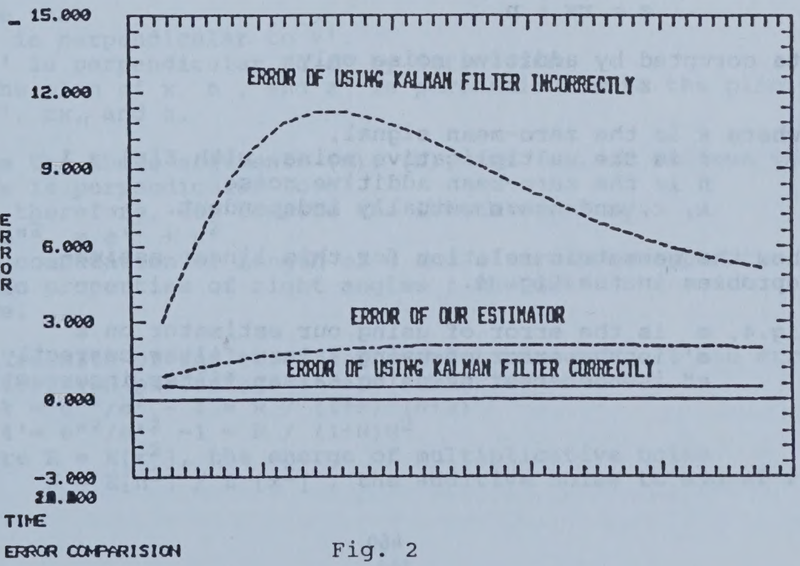


Fig. 2

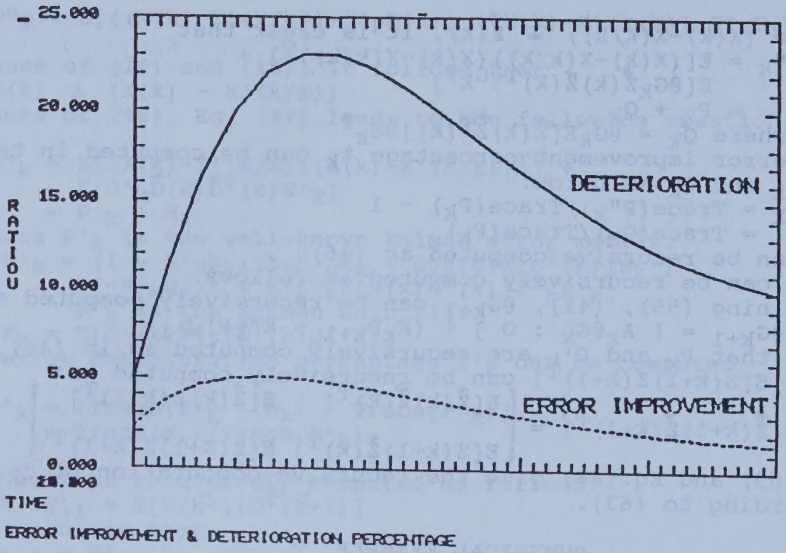


Fig. 3

GEOMETRICAL INTERPRETATION

To understand the role of multiplicative noise, we give an interpretation from geometrical point of view, which will give us a clear picture for the linear estimation. Here, we consider a simple case, one single estimation using one data. The signal, noises, and data all scalar random variables in this case.

.Data corrupted by multiplicative and additive noise:

$$z = rx + n$$

.Data corupted by additive noise only:

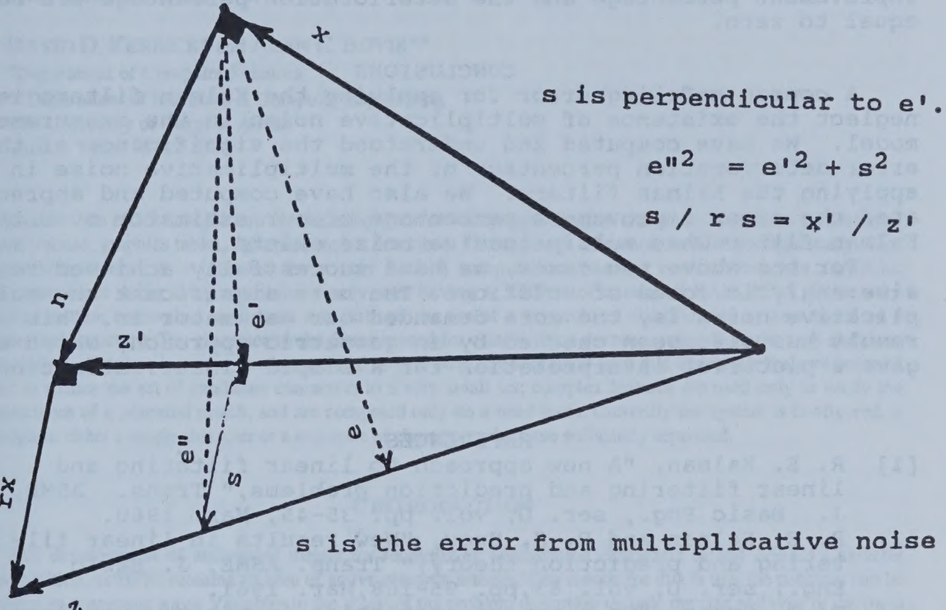
$$z' = x + n$$

where x is the zero-mean signal,
r is the multiplicative noise, with $E[r] = 1$
n is the zero-mean additive noise
x, r, and n are mutually independent.

We show the geometric relation for this linear estimation problem in the Fig. 4.

In Fig.4, e is the error of using our estimator on z
e' is the error of using Kalman filter correctly on z'
e" is the error of using Kalman filter incorrectly on z

The geometric relation is shown in the diagram below:



x
s is perpendicular to e'.
$e''^2 = e'^2 + s^2$
$s / r s = x' / z'$
n
e'
z'
e"
s
e
rx
s is the error from multiplicative noise
z

Fig.4

As shown in the diagram,

$$e'' = e' + s$$

where s is the effect from multiplicative noise.
Note that the length of s can be computed from the follow equation

$$s / r s = x' / z'$$

Note that
(a) s is perpendicular to z'.
(b) e' is perpendicular to z' by definition.
(c) the plan of x, n , and z' is perpendicular to the plan of z', rx, and z.

From the above arguments (a), (b), and (c), it follows that s is perpendicular to e'.
We, therefore, can compute the length of e" by

$$e''^2 = e'^2 + s^2$$

The computation of length of e and e' are basee upon the geometric properties of right angles ; the details are not shown here.

The results of the error inprovement percentage % and error deterioration percentage %' are summarized below

$$\% = e''^2/e^2 - 1 = R / (1+N)^2(N+2)$$

$$\%' = e''^2/e'^2 - 1 = R / (1+N)N^2$$

where $R = E[r^2]$, the energe of multiplicative noise
$N = E[n^2] / E[x^2]$, the additive noise to signal ratio

Note that If there is no multiplicative noise (R = 0), the inprovement percentage and the deterioration percentage are both equal to zero.

CONCLUSIONS

A common modeling error for applying the Kalman filters is to neglect the existence of multiplicative noise in the measurement model. We have computed and understood the significance of the error deterioration percentage of the multiplicative noise in applying the Kalman filters. We also have computed and appreciated the error improvement percentage of our estimator over the Kalman filter when multiplicative noise exists.

For the above two cases, we have successfully achieved recursive analytic forms of solutions. The more significant the multiplicative noise is, the more demanded our estimator is. This result has also been checked by an geometric approach, which we gave a pictorial interpretation for a simple linear estimation.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and linear filtering and prediction problems," Trans. ASME, J. Basic Eng., ser. D, vol. pp. 35-45, Mar. 1960.

[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," Trans. ASME, J. Basic Eng., ser. D, vol. 83,pp. 95-108,Mar. 1961.

[3] T. T. Soong, "On a priori statistics in minimum variance estimation problems," Trans. ASME, J. Basic Eng., Series 87D, pp. 109-112. Mar. 1965

[4] H. Heffes, "The effect of erroneous models on the Kalman filter response," IEEE Trans. Automat. Contr., vol. AC-11, PP. 541-543, July 1966

[5] T. Nishimura, "On the a priori information in sequential estimation problems," IEEE Trans. Automat. Contr., vol. AC-11, pp. 197-204, Apr. 1966.

[6] B. Friedland,"Treatment of bias in recursive filtering," IEEE Trans. Automat, Contr. vol. AC-14, pp. 359-367, Aug. 1969.

[7] H. R. Ramer, Statistical Communication Theory and Applications: Englewood Cliffs, N. J.: Prentice-Hall, 1969

[8] B.S. Chow and W. P. Birkemeier, "A new structure of linear recursive estimator," Proceedings 1987 American Control Conference, pp791-796.

[9] B. S. Chow and W. P. Birkemeier, "A recursive estimator using data in blocks as applied to time invariant systems with multiplicative noise," 1987 proceedings IEEE International Symposium on Circuits and Systems. pp. 158-162.

[10] B. S. Chow and W. P. Birkemeier, "New results of state estimation under uncertain observation," 1986 Proceedings Allerton Annual Conference. University of Illinois, Urbana, pp. 125-134, also modified in review to IEEE Trans. on Information Theory.

EFFICIENT RECOGNITION OF HANDPRINTED CHARACTERS FROM TABLET-BASED INPUT

DAVID D. KERRICK* and ALAN C. BOVIK**
*Department of Computer Sciences
**Department of Electrical & Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084

Abstract -- An efficient algorithm for the recognition of handprinted English characters entered on a tablet-type input (mouse, graphics tablet, light pen, etc.) is described. The approach taken here is different from many other algorithms in that it is intended for a very limited computational/storage environment, viz., an 8-bit microprocessor using integer arithmetic only, and having ≤ 32kbyte of combined RAM/ROM. These very limited system constraints have been met through the use of a heirarchical classification system based on a binary tree structure. The character/stroke features used for classification purposes range from the very simple to the slightly complex; however, the heirarchical control structure is defined such that the simple features are used first to reduce the set of candidate characters to a very small set; complex features are used only to verify the correctness of a potential match, and are computed only on a need basis. Currently the system is configured to recognize either a single character or a sequence of characters which are sufficently separated.

I. INTRODUCTION

The development of automated means for recognizing handprinted characters (Hand Printed Character Recognition, or HPR) remains an area of active research interest. One reason for this is that the problem can be defined in numerous ways. Variables in the scope of the problem definition include the size and type of the input character set, the character input medium (e.g., digital image, tablet, etc.), computing requirements, matching accuracy, etc. Each problem definition lends itself to many different algorithmic approaches for recognizing characters. Some of the control structures proposed for HPR include sequential, hierarchical, and tree-based methods.(1) Various character features have been used for describing and classifying input character sets. Feature descriptions have been based on both globally and locally computed features,(2) statistical measures,(3) and Fourier descriptors,(4) among others. The computing resources used to solve the problem have ranged from large mainframe computers to small microcomputers.

The approach taken in this paper combines elements from several of the above methods to give a unique system for solving the HPR problem. Using very modest computation resources (8-bit microprocessor, ≤ 32 kbyte of memory) the algorithm is currently able to classify a large character set (69 characters) with a high degree of accuracy in a short amount of time (~ 50 ms). The character set consists of the upper and lower case English block letter alphabet, the Roman numerals 0-9, the basic math operators (+, -, *, /), and a few special characters. The control structure is hierarchical in nature, incorporating a simple binary decision tree for fast classification. The algorithm uses a combination of globally and locally defined character features which rapidly reduce the candidate character matches to a very small set. The hierarchical control strategy avoids the use of computationally expensive measurements until only a very few (often just one) candidate matches remain. Moreover, the more expensive measurements are only computed on a need basis, viz., to verify the correctness of a candidate match or small set of matches.

II. SYSTEM OVERVIEW

Here we discuss three important and interrelated components of any effective HPR system, and how these components were defined in this system: (1) system constraints, including input medium; (2) choice of features; and (3) control structure. The particular combination of these three components is what makes the HPR system described here unique.

2.1 *System constraints*

The most important constraint used in the design of the system described here is the choice of a very limited computational resource. Specifically, the algorithm was constrained to operate in a small (8-bit) microprocessor having modest memory capabilities (≤ 32kb) and using only integer arithmetic. Another important consideration is the desired speed and accuracy of the system. Since the proposed system is intended for on-line classification, fast processing is essential (≤ 50ms). In addition, a high degree of accuracy and flexibility is desired to allow for the possible recognition of characters having multiple renditions. Perhaps the most influential factor in the design of an HPR algorithm is the choice of input medium. The system described here operates using a stroke-based input, viz., vectors of x-y coordinate pairs obtained from, e.g., a light pen, resistive pad or other tablet, or a mouse. An obvious advantage of this medium is that character preprocessing becomes superfluous, viz., there is no need for character thinning or completion prior to processing (as is the case with image-based systems). The characters are assumed to be entered with sufficient separation so that there is no ambiguity in associating each stroke with the correct character; thus, the algorithm is able to accomplish simple parsing of character streams and accurate recognition of each character. Since the strokes comprising the characters are easily and compactly stored, the system lends itself to concurrent classification and character input: some features can be computed as the character is entered. Aside from the advantages offered by the input medium, the choice of a stroke-based input has a profound influence on the choice of features used, and on the design of the system as a whole.

2.2 *Feature selection*

An "optimal" set of character features for an HPR system should:

(a) Fully capture the "essences" of the character set;

(b) Require minimum processing time for extraction from the input character renditions.

In an effort to define an "optimal" set of character features, we first identified those primitive features of hand printed characters containing significant information. These primitives can be classified into two general types: (1) Locally defined features such as discontinuities (*endpoints*), points of high curvature (*corners*), and joins of three or more strokes (*tees*), and (2) Globally defined features such as the *relative* locations of the locally defined features within a given character, character aspect ratio, and gross descriptions of the shape or curvature of a stroke or sequence of character strokes.

The features chosen for this system achieve goal (a) by combining both local and global information into each feature. Additionally, different features embody the character at different levels of abstraction. Some features are based mainly on local character properties while others capture mainly global properties, the local aspects of the feature being implicit. This choice of features is particularly advantageous due to the input medium. The simplest features are trivially extracted from an input sequence of character rendition strokes entered on a tablet. However, the considerable information content of the features used allows for a rapid reduction in the set of possible character matches. The extraction of high-level (and more computationally expensive) features is postponed until after the low-level features are identified and used to reduce the candidate set. Pre-classification using these simple features results in a small number of possible character matches. Extraction of high-level features occurs on a limited scale and is facilitated and directed by the small set of remaining candidate character matches. Therefore, the second goal (b) is achieved through both the choice of features and the use of a control structure which exploits easily obtainable information to limit and direct the extraction of higher level features. Through this control structure we are able to define a fast, accurate, and easily understood HPR system. Most of the features that are used in the HPR system are not "new"; many of them have been used in various combinations in other HPR, or more generally, character recognition systems. The efficacy of the HPR system described here lies in the specific combination of features, the system hierarchy and the control structure used in defining the system.

2.3 *System control structure*

The overall control structure employed is hierarchical; however, the different hierarchies employ different control structures themselves. The choice of control structure was dictated by the choice of features, and was designed to exploit them in such a way as to require minimum processing time for each character rendition. The three hierarchies of control are based on three levels of feature abstraction. The lowest level features are extracted first and are subsequently used to reduce the number of possible character matches via a binary decision tree. Each node in the tree tests for the presence or absence of a feature, with classification proceeding along appropriate subtrees accordingly. This fast and efficient classification method quickly reduces possible character matches and is largely unaffected by the size of the character set (see Section 3). The next level of features are slightly more complex, yet the classification method is very similar. For reasons made clear later, only the absence of a feature is used as a classification criteria: a binary decision tree is used as a control structure but only the absence of features is used as a node test. The highest-level features or stroke properties are used after the rendition is parsed according to the expected properties of the remaining candidate character matches. *If* the rendition can be parsed according to these possible matches, *only then* are individual stroke properties computed and compared to stored character and stroke property models; otherwise the entered character rendition is rejected as being meaningless.

III. FEATURE DEFINITIONS & FEATURE EXTRACTION

The definition of the set of features to be used for classification purposes is concommitant to the development of any HPR system. Accordingly, a description of the features and the extraction methods used to obtain them would be incomplete without justifying the choice of features. The features used in the HPR system described here can be grouped into three levels of complexity or abstraction.

3.1 *Low-level features*

The lowest level features are very simply defined, yet provide much of the power of the proposed HPR algorithm due to the combination of local and global structural character information contained in them. The presence of *endpoints* (isolated stroke terminations) and/or *tees* (where two stroke segments join) comprise the local information, as illustrated below in Fig. 1. Once these simple features have been located, the relative location of the endpoints/tees with respect to the character as a whole are used to determine the subset of the character set which may coincide with the rendition.

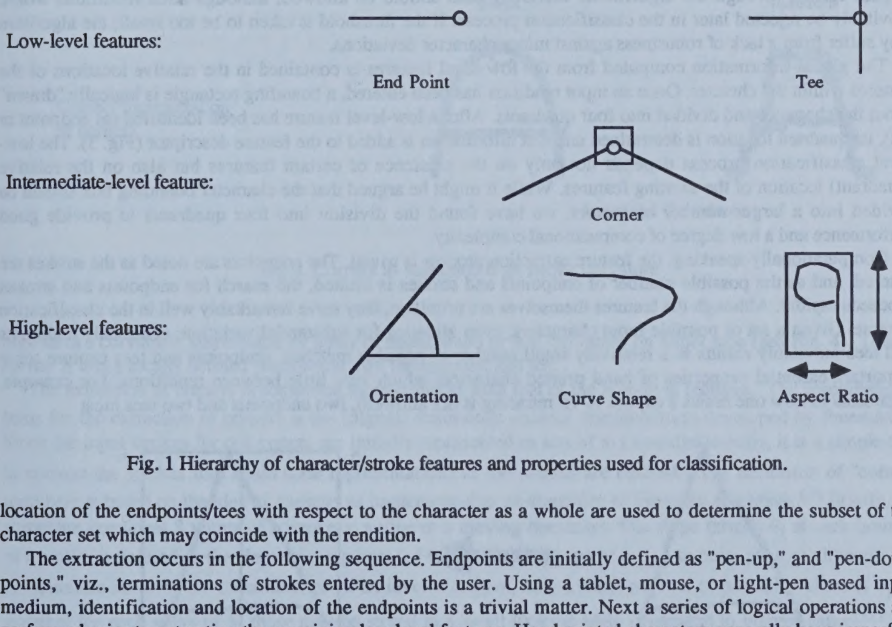


Fig. 1 Hierarchy of character/stroke features and properties used for classification.

The extraction occurs in the following sequence. Endpoints are initially defined as "pen-up," and "pen-down points," viz., terminations of strokes entered by the user. Using a tablet, mouse, or light-pen based input medium, identification and location of the endpoints is a trivial matter. Next a series of logical operations are performed prior to extracting the remaining low level features. Hand printed characters generally have imperfect stroke shapes, connectivity or stroke positioning. It is often the case that a stroke may be entered in a non-continuous manner, giving rise to breaks or gaps in what is intended to be a single stroke or a connected series of strokes. Thus we attempt to recognize "imperfect" features. Prior to classifying the input rendition using the low-level features, all endpoints lying within a threshold distance of one other are "logically" joined, removing these endpoints from consideration as features (Fig. 2). This results in a single stroke having only two endpoints, where there had originally been two strokes and four endpoints. The threshold distance used for joining endpoints is directly related to the overall size of the input character rendition.[1] Each remaining

---

[1] The threshold is calculated as an integer fraction of the largest (width or height) dimension of the character. Currently, the threshold is taken to be an eighth of the maximum character dimension.

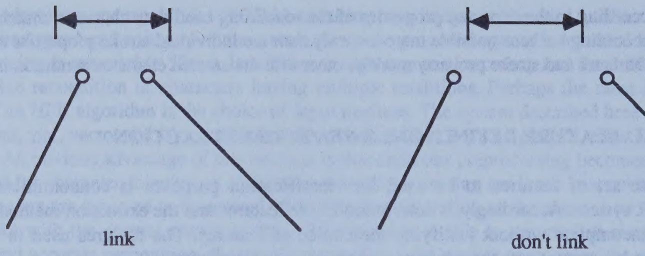


Fig. 2. Logical linking of endpoints based on proximity

endpoint is then checked for proximity to a stroke; again, a threshold is used to constrain the search area.[2] If an endpoint is sufficiently "close" to a stroke, the endpoint and stroke are logically joined to form a tee. This logical linking only involves noting the existence of the new feature and the stroke and endpoint involved. The endpoint is no longer considered an endpoint but rather as a tee. The choice of the thresholds (which are equal) used for computing endpoint/endpoint and endpoint/stroke proximity remains largely *ad hoc* by necessity. If a very large threshold is used, it is possible that very ill-formed or nonsensical character renditions may be allowed to pass further through the algorithmic hierarchy than should be allowed, although such renditions would inevitably be rejected later in the classification process. If the threshold is taken to be too small, the algorithm may suffer from a lack of robustness against minor character deviations.

The global information computed from the low-level features is contained in the relative locations of the features within the chracter. Once an input rendition has been entered, a bounding rectangle is logically "drawn" about the character and divided into four quadrants. After a low-level feature has been identified (as endpoint or tee), its quadrant location is determined and this information is added to the feature descriptor (Fig. 3). The low-level classification process depends not only on the existence of certain features but also on the relative (quadrant) location of the existing features. While it might be argued that the character bounding box should be divided into a larger number of sectors, we have found the division into four quadrants to provide good performance and a low degree of computational complexity.

Computationally speaking, the feature extraction process is trivial. The endpoints are noted as the strokes are entered, and as the possible number of endpoints and strokes is limited, the search for endpoints and strokes proceeds rapidly. Although the features themselves are primitive, they serve remarkably well in the classification scheme. Given a set of possible input characters, even allowing for substantial variation, using only endpoints and tees inevitably results in a relatively small number of possible matches. Endpoints and tees capture some important essential properties of hand printed characters which vary little between renditions. For example, regardless of how one draws a capital 'A' (if retracing is not allowed), two endpoints and two tees must invariably be rendered, the endpoints being constrained to the lower left and the lower right quadrant. Although it may be easily argued that a complete set of renditions of character 'A' should allow for slanted versions of the character (so that the endpoints may both lie in one of the lower quadrants), this case may be either treated

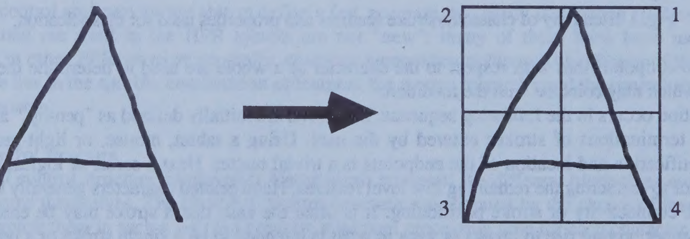


Fig.3. Quadrants used in feature location. The quadrants are formed from the bounding box of the rendition.

---

[2] This is a simple and rapid process since the number of endpoints for a character is usually two and no more than four.

simply as a separate rendition; however, it might also be preferable to treat this as an ill-formed, and hence, unacceptable character.

The system we describe here is configured to operate in an highly constrained computational environment, when compared with other HPR algorithms. As such, the algorithm does suffer from some limitations; chief among these is that the user is not allowed to "retrace" the strokes comprising the characters, except when such a retracing is a natural means for entering the character. The letters for which retracing is allowed include only certain renditions of the lower-case letters 'a,' 'g,' 'h,' 'm,' 'n,' 'q,' 'r,' 'u,' and 'y'.

3.2 *Intermediate level features*

The single intermediate level feature is the corner (Fig. 1). Although the corner is a locally defined feature, as with the low-level features the location of each corner is used as a source of global information which is assigned to each identified corner. "Corner" is a rather imprecise term; by corner we mean an area of *sustained* high stroke curvature, surrounded by areas of unsustained or low curvature on either side (see Fig. 4). Thus, a *corner* is less a locally defined phenomena than the endpoint and tee.

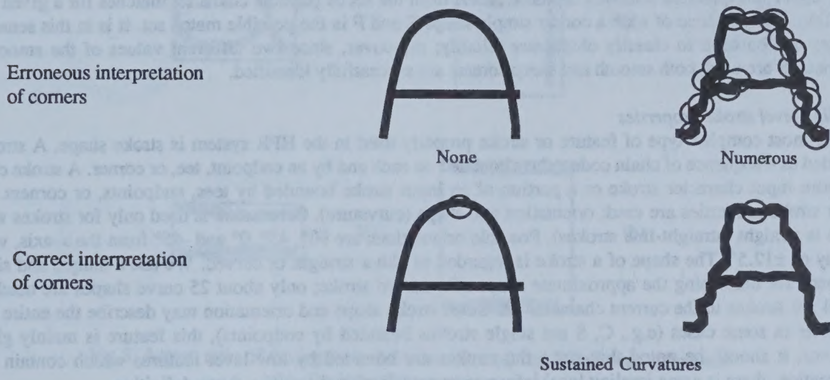

Erroneous interpretation of corners
None
Numerous
Correct interpretation of corners
Sustained Curvatures

Fig.4. Corners as sustained high stroke curvature

The extraction of corners is somewhat more complex than that of the low-level features. The computational basis for the extraction of corners is the (digital) chain-code contour representation developed by Freeman.(6) Since the input strokes for our system are initially represented as sets of x-y coordinate pairs, it is a simple task to convert the strokes into chain code representations as the strokes are entered.[3] The definition of "corner" used here is based on the idea of *corneity* as implemented in an algorithm of Freeman and Davis.(7) Briefly, the algorithm considers S points of a stroke at a time in a moving operation. The slope (angle) θ_i at each point p_i of a stroke is defined as the slope of a segment (of length S) joining point $p_{i-S/2}$ to point $p_{i+S/2}$; although this computation involves the determination of an inverse tangent, only a small number of possible values of θ_i are possibel for each value of S; these may be stored in a small look-up table (truncated to integer value).(7) The change in slope at each point p_i in the stroke is given by the incremental curvature $\delta_i = \theta_{i-1} - \theta_{i+1}$. A corner is defined as a collection of contiguous points on a stroke, each point having an incremental curvature δ greater than a threshold (which can analytically determined as a function of S; see Ref. 7).[4] Furthermore, the set of stroke points defining the corner must have some minimum cardinality, which also can be determined (as a function of S).(7) The actual location of the corner is taken to be that point within the region having the largest incremental curvature δ. The parameter S controls the amount of smoothing of the particular character stroke being processed. In an effort to increase the reliability of the system, the HPR algorithm incorporates two values of S in the corner finding stage; thus, by definition of an appropriate set of thresholds, the algorithm finds both abrupt corners and more gradual but more sustained corners. As with the thresholds used for low-level feature extraction, the values of S used by the algorithm are related to the overall input character rendition size. Since the algorithm results in a measure of "corneity" for all points, the algorithm identifies not only the points of high discontinuity but also their degree of "corneity." The algorithm also allows for the detection of corners occurring at logical joins of endpoints (see Section 2) in a very simple manner; simply, two strokes "joined" at endpoints are treated as one stroke in forming the chain code representation, viz., the chain code is simply continued from one stroke to the next without including the link between endpoints.

---

[3] The conversion process is in fact linear.(7)

[4] The threshold used in this algorithm is $\tan^{-1}[1/(3+s)]$.

The justification for the use of corners as primitive character features is similar to that for the low-level features. Each stage of the corner computation is of linear complexity. Additionally, through the use of look-up tables (of very reasonable size), all computation can be accomplished using integer arithmetic. The corners, however, are somewhat less essential to the actual character description. Renditions of characters often contain many spurious corners. These will be (unavoidably) identified as potentially meaningful features. However, this does not present a problem since it is the *absence* of corners and not their presence which is used by the classification algorithm (except in a very limited number of special cases). For example, the absence of a corner in the upper left quadrant removes capital F and E from the set of possible character matches for a given input rendition. The presence of such a corner simply keeps E and F in the possible match set. It is in this sense that corners can be used to classify characters reliably; moreover, since two different values of the smoothing parameter S are used, both smooth and sharp corners are successfully identified.

3.3 *High level stroke properties*

The most complex type of feature or stroke property used in the HPR system is stroke shape. A stroke is recorded as a sequence of chain code values bounded on each end by an endpoint, tee, or corner. A stroke can be an entire input character stroke or a portion of an input stroke bounded by tees, endpoints, or corners. Two major stroke properties are used; orientation and shape (curvature). Orientation is used only for strokes whose shape is straight (straight-line strokes). Possible orientations are 90°, 45°, 0° and -45° from the x-axis, with a leeway of ±12.5°. The shape of a stroke is regarded as either straight or curved. We use a unique and simple approach for describing the approximate shape of a curved stroke; only about 25 curve shapes are needed to model the strokes in the current character set. Since stroke shape and orientation may describe the entire input character in some cases (e.g., C, S are single strokes bounded by endpoints), this feature is mainly global, however, it should be noted that since the strokes are bounded by low-level features which contain local information, there is some implicit local information contained in the stroke shape definition.

Although the shape descriptor is the most computationally expensive stroke feature or property computed in the algorithm, the orientation of a (straight) stroke is trivial to compute. Given that a stroke is "straight," it is a simple task to determine its orientation by calculating its slope from the chain-code representation.

Before detailing the extraction of stroke shape, we briefly discuss how these features are used. Unlike the low-level and intermediate features are were identified and then used in classification, the higher level (and more computationally expensive) shape properties are extracted strictly on a *need basis*. That is, stroke shape is computed only if a candidate character match contains a stroke of a specific shape.

The extraction process is defined as follows: given a stroke (input line segment bounded by low-level features) the shape is computed in one of two ways depending on whether a straight or curved stroke is expected. To determine whether a stroke is straight we reuse the algorithm from the intermediate level features. If no corner above a certain threshold magnitude exists on the stroke, then the stroke is straight, otherwise it is not straight. If a curved stroke shape is expected, the algorithm verifies whether the stroke has the appropriate shape. To accomplish this, a very simple, easily computed, and effective "shape descriptor" is defined, as follows: each (curved) stroke is divided into four segments of equal length. The chain code for each segment is determined and the average chain code value of each is computed.[5] Thus, a four-digit number is obtained which roughly describes the shape of the stroke. The four-digit number is then compared to a similar number associated with the idealized curve-shape. If any of the four values differ by more than one from the ideal, the stroke shape is regarded as not having the expected shape. This descriptor has proved to be remarkably robust in capturing the shapes of hand drawn strokes.

Although the reliability of the shape descriptor may seem surprising, the constraints on how descriptors are computed and used contribute to their reliability. As stated earlier, there is implicit global information in the low-level features which bound the strokes. Moreover, at a coarse resolution level (such as that used in computing the four-digit curve shape descriptor) only a finite number of shapes are possible. Finally, the strokes comprising the input rendition have a limited variation in shape. Although the stroke descriptors are relatively expensive to compute, they involve (at worst) the averaging of chain code values.

---

[5] Using modulo-8 arithmetic.

3.4 *Other descriptors*

Using the stroke properties described in the preceding, the HPR algorithm generates a single character match for nearly all input renditions. However, in some cases involving highly similar characters, there may be remaining ambiguities which the stroke properties cannot resolve. One means for resolving this ambiguity is by using character aspect ratios. The aspect ratio is useful since it is required only to resolve a few cases of very limited ambiguity: given two possible matches for an input rendition, determine the correct match using the heighth/width aspect ratio of the input rendition. The characters pairs which require the aspect ratio measurements include only 'a' vs. '9' and 'h' vs. 'n'. In other cases, the algorithm requires the aspect ratio to fall within a limited range, in order to eliminate ill-formed renditions. These only include the characters '+' and '='. Examples are given below in Fig. 5.

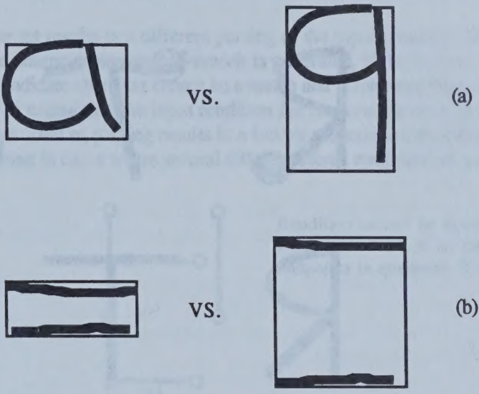


Fig.5. Aspect ratio of the input rendition used to: (a) resolve similar characters
(b) eliminate ill-formed characters.

A final source of ambiguity arises for characters having similar **model-chars** and aspect ratios. This type of ambiguity arises primarily because the existence of a corner is not used by the binary decision tree to eliminate candidate characters; only the absence of a corner is used to reduce the candidate set size (Section 3.2). However, in two cases, the algorithm does use the presence of a corner to differentiate between these highly similar characters. Thus, an input rendition which matches models for both '5' and 's' is classified as a '5', while a character which matches the models for both '4' and '9' is regarded as a '4'; viz., the more restricted model ***requiring*** the existence of certain corners is accepted. Although this approach may seem somewhat *ad hoc*, in the absence of contextual information it may be impossible for even a human observer to arrive at a conclusive classification.

Every character in the current set can be classified by the HPR algorithm using the stroke features/properties described thus far with the exception of two: the lower cases letters 'i' and 'j'. In defining the stroke types, we have not accounted for the possible renditions of the dots over these characters; the problem is complicated by the fact that different users may render the dot in many different forms. We have chosen to handle the dots by constraining the user only to render the dot over a small area sufficiently separated from the remainder of the stroke; thus, a stroke is identified as a dot if it is both isolated from the rest of the character (exceeds the proximity measure used in defining joins and tees), and if it is limited in spatial extent (is no larger than twice the same proximity measure along either dimension).

IV. CLASSIFICATION AND CONTROL STRUCTURE

The classification of characters in the system is accomplished in three stages. First, a binary decision tree is defined which uses the low-level and intermediate-level features to reduce the number of possible character matches to a small candidate set. Second, the candidate character matches remaining are used to parse the input rendition into a set of stroke segments. Finally, for those characters for which the parsing stage is successful, individual stroke properties are compared to a model character.

4.1 *Binary decision tree*

The goal of this stage of classification (*Stage 1 Classification*) is simple: reduce the number of candidate character matches to a small set. The binary decision tree was chosen as a control structure both for its efficiency and its computational simplicity. Each node in the tree tests the input rendition for either the absence or presence of a particular feature in a given quadrant: a sequence of binary decisions. A character or set of characters is defined by a path in the tree from root to leaf node; the leaf node being the end of Stage 1 Classification.

Classification using low-level features is straightforward; the presence or absence of a feature is used as a node test. However, due to the nature of hand drawn characters, classification using corners is necessarily different. If a character is drawn carelessly or in a highly stylized fashion, extraneous corners (corners not essential to the character) may be detected (Fig. 6). Detection of these corners cannot be avoided, hence, classification using corners only involves testing for the presence of *necessary* corners; the extraneous corners are effectively ignored.

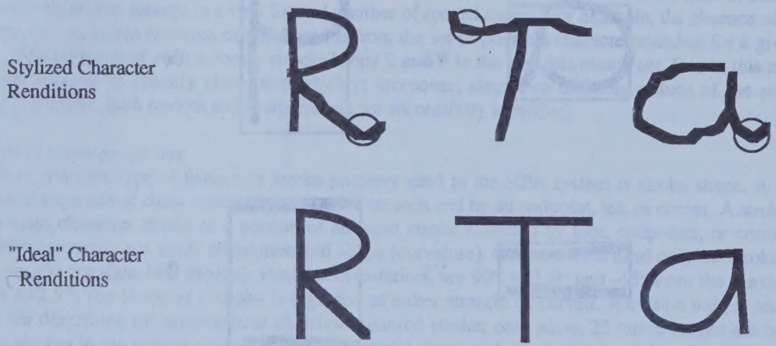


Fig.6. Extraneous corners in highly stylized or poorly drawn characters

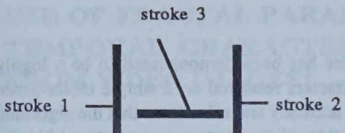
One important advantage of the binary decision tree control structure is that alternate character definitions need only be included as additional search paths and final nodes in the tree. Severely slanted or skewed character renditions will have different relative feature positions yet are easily accommodated by including an appropriate character definition in the tree. The power of the approach is intrinsic in the binary decision tree structure. The number of decisions required to classify a character as one of a set of N possible characters is approximately $\log_2 N$. Hence, doubling the size of the character set (e.g., either by including alternate character definitions or by extending the existing character set) only increases the search path by approximately one node (on the average). Perhaps most importantly, no extensive matching need be performed: all decisions are simple binary decisions, so that the time to traverse a search path is very small.

4.2 *Character parsing*

The completion of Stage 1 Classification results in a small set of candidate character matches. Stage 3 Classification involves comparing the properties of individual strokes in the input rendition to properties expected in each for these candidate matches. Character parsing is a means of organizing the input character rendition into strokes in order to make these comparisons. Before parsing, information on the input rendition consists of (1) the original input stroke vectors, and (2) the features, which are records of the stroke and the position of each feature identified.

Two data structures are used in both Stage 2 and Stage 3 Classification. The first is a **model-char**. A **model-char** is stored for each member of the possible input character set. The **model-char** consists of a set of stroke labels, each label having associated stroke properties (see Fig. 7). The second data structure is a **possible-match** which is generated following the successful parsing of an input rendition. The **possible-match** has a name field to identify the possible candidate match it represents. The **possible-match** also contains a set of stroke segment starting and ending points. Each start and end point pair is intended to enclose one stroke segment. Given the **possible-match** and the input rendition, the rendition can be viewed (and processed) as a set of stroke segments.

Thus, *Stage 2 Classification* proceeds as follows. The parsing algorithm uses the features to identify and label strokes according to a possible character match. Comparison of the input rendition to each member of the

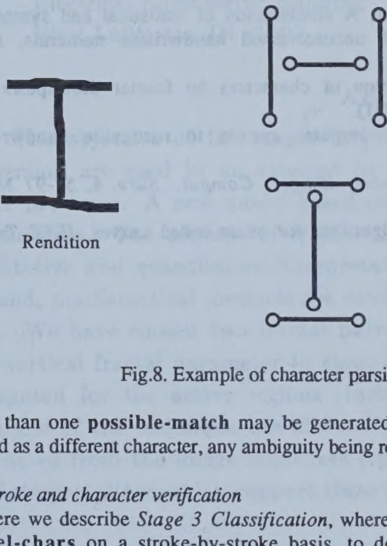


```
Model [8] .NumStokes = 3                (* 3 strokes *)

Model [8] .Stroke[1] .Shape = Str       (* All straight *)
Model [8] .Stroke[2] .Shape = Str
Model [8] .Stroke[3] .Shape = Str

Model [8] .Stroke[1] .Orient = 90       (* Orientations *)
Model [8] .Stroke[2] .Orient = 90       (* from x-axis *)
Model [8] .Stroke[3] .Orient = 0
```

Fig.7. Data Structure **MODEL-CHAR** for character **H**

(small) candidate character set results in a different parsing of the input rendition. If the rendition can be parsed according to the candidate character, a **possible-match** is generated. If the parsing cannot be performed on the input rendition, then that candidate character cannot be a match and is removed from the candidate character set.

Therefore, in addition to organizing the input rendition for final verification based on stroke shape, parsing has an important side effect. That is, parsing results in a further reduction of the candidate character set (see Fig. 8). It should also be noted that in cases where several different stored renditions of a single character are possible, more than one **possible-match** may be generated. In this case, the multiple **possible-matches** are each treated as a different character, any ambiguity being resolved in the final verification step.



Rendition cannot be matched to this model character - there is no stroke connecting the endpoints in quadrants II and III (or I and IV)

Rendition can be matched to this model character.

Fig.8. Example of character parsing reducing the candidate character set.

4.3 *Stroke and character verification*

Here we describe *Stage 3 Classification*, where the set of **possible-matches** are compared to the set of **model-chars** on a stroke-by-stroke basis, to determine which of the candidates, if any, sufficiently approximates the input rendition. The verification stage of the classification process has two main goals: (1) identify which character/characters have the appropriate stroke properties to match the input rendition, and (2) resolve any remaining ambiguity in the final identification of the rendition.

Verification using stroke properties uses both the model stored by the system and the **possible-matches** constructed by Stage 2 Classification. Each stroke segment defined by the **possible-match** (i.e. the segment enclosed by a start point-end point pair) corresponds to a stroke label and is property stored in a **model-char**. The properties of the stroke segments are computed (see Section 3.3) and compared to the expected properties of the labelled strokes. If all stroke segment properties match the expected properties, than the rendition is regarded as a match to the **model-char**. If any stroke segment fails to match the corresponding stroke in the **model-char**, that character is removed from the candidate character set, and a new candidate character is tried. This process is repeated until the candidate character set is exhausted.

V. SUMMARY

The Handprinted Character Recognition algorithm described here has been demonstrated to be a highly effective and efficient means for classifying handprinted block characters rendered on a simple tablet-based medium. The extreme simplicity of the algorithm belies the degree of accuracy and robustness that the algorithm achieves; however, this simplicity was motivated by the configuration of the system to operate within the confines of a strict computational framework. However, the application scope of the algorithm is somewhat limited in its present form. The algorithm could be improved by either (a) increasing the size of the character set to allow for a larger variety of stylized or ill-formed renditions, or by (b) incorporating contextual information into the classification process to allow for classification based on cooperative examination of surrounding characters. Both approaches would require an increase in both computational and storage capacity.

REFERENCES

1. M. Shridhar and A. Badreldin, A tree classification algorithm for hand written character recognition, *Proc. 7th IJCPR*, 615-618 (1984).
2. W. Verschueren et. al., Structural recognition of handwritten numerals, *IEEE Trans. Pattern Anal. Mach. Intell.*, 760-762 (1984).
3. B. Duerr, W. Haettich, H. Troph and G. Winkler, A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals, *Patt. Recog.* **12**, 189-199 (1980).
4. M.T.Y. Lai and C.Y. Suen, Automatic recognition of characters by fourier descriptors and boundary line encodings, *Patt. Recog.* **14**, 383-393 (1981).
5. G.Y. Tang, P. Tzeng, and H. Hau, A microcomputer system to recognize handwritten numerals using a syntactic-statistical approach.
6. H. Freeman, Computer processing of line-drawing images, *Comput. Surv.* **6**, 57-97 March (1974).
7. H. Freeman and L.S. Davis, A corner-finding algorithm for chain-coded curves, *IEEE Trans. Comput.*, 287-303 (1977).

USE OF FRACTAL PARAMETERS TO DETERMINE LOCAL SPATIO-TEMPORAL CHARACTERISTICS OF A TURBULENT POLLUTANT FROM VIDEO IMAGES†

AMAR AIT-KHEDDACHE
North Carolina State University
Electrical and Computer Engineering Department
Campus Box 7911
Raleigh, NC 27695-7911

SARAH A. RAJALA
Purdue University
School of Electrical Engineering
Electrical Engineering Building
West Lafayette, IN 47907

ABSTRACT

The objective of this paper is twofold. First, visual images of smoke plume dispersion are used in an attempt to develop techniques for describing important wake processes. A new model based on fractal concepts is developed to analyze the smoke data. The concept of fractals is introduced for the purpose of giving some qualitative and quantitative interpretations to the transient flows of the pollutant. Second, mathematical methods are developed for making the concept useful in practice. We have chosen two fractal parameters, the horizontal fractal parameter and the vertical fractal parameter to characterize the image data. These parameters are computed for the active regions (turbulent regions) of the phenomenon (smoke). Analysis of the technique has allowed us to determine which smoke features can be extracted from the image sequences (spatio-temporal characteristics). Some statistical interpretations which support these results are reported.

BACKGROUND

Understanding the process of dispersion in the atmosphere is central to quantitative assessment of the impact of the release of pollutants. In addition, there is a continuing interest in improving the design of factories emitting pollutants (e.g., coal-burning power plants) to avoid the effects of down-draughts and to examine the flow characteristics near buildings or other obstacles.

Meteorological measurements of the effect of buildings on atmospheric turbulence and the dispersion of pollutants have been investigated experimentally by Robins and Castro [1], numerically by Frost, et al. [2], and analytically by Hunt, et al. [3]. These studies are primarily based on the measurement and characterization of the structure that has been observed through the development of a multi-point

---

† This research was supported by EPA project number CR813368-01.

measurement methodology that reflects both the presence of coherent structures and their random occurrences. The measurements include: average speed, spatial distribution of the pollutant, quantity of flow and direction.

Another class of measurement techniques that has been used to enhance the understanding of the atmospheric dispersion of hazardous air pollution is based on remote sensing techniques. The importance of remote sensing techniques arises from the fact that the data can be collected both at ground level and higher levels in the atmosphere. Data collected at higher levels in the atmosphere provide researchers with information about both the physico-chemical reactions and photo-chemical transformations that the pollutants experience far away from their origins. The remote sensing methodology includes: correlation spectroscopy [4-5], single wavelength lidar and differential lidar (DIAL) [6-8], sound detection and ranging (SODAR) [9], and airborne sensors.

A new approach for measuring and analyzing the spatio-temporal distribution of pollutants is presented in this paper. The objective of this approach is to combine the theory of fractals and image analysis techniques to investigate the dispersion of pollutants. A theoretical justification is given and mathematical methods are developed to make the concept useful in practice. Furthermore, an attempt is made to understand and quantify the complex aspects of the turbulent motion.

FRACTAL CONCEPTS

Fractal set theory was first mentioned a century ago in a correspondence between Cantor and Dedekind who discovered some mathematical structures that were not adequately described by Euclid's or Newton's models. Two decades ago, Mandelbrot [10-11] pioneered the use of fractal models and paved the way to solve some previously unsolved problems in economics and the hard sciences.

The term "fractal" was derived from the Latin adjective *fractus*, meaning broken or fragmented, by Mandelbrot. Furthermore, he proposed the definition of a fractal as a set (subset of R^n) with its Hausdorff-Besicovich dimension strictly greater than its topological dimension [11]. The topological dimension is an integer representing the number of different orientations within a set. However, Mandelbrot pointed out that his definition of a fractal seemed incomplete because it excluded a number of irregular sets which should be categorized as fractals. Moreover, fractals represent the large number of complicated objects which are statistically self-similar. Technically, the self-similarity property may be applied to either bounded or unbounded sets, as long as each piece of the shape is geometrically similar to the whole.

Why do we want to use fractals? First, many natural objects are fractal. Knowing that an object is fractal gives one a predictive power if the dynamical fractal parameters characterizing the object are known. Second, a qualitative relationship exists between the dynamical fractal parameter and other information about a natural phenomenon. As a result a large number of scientists from different backgrounds have begun to use fractals.

A large number of methods have been used to generate fractal or near fractal sets. Wilson [12] used cellular automata to obtain certain fractals. Ogden [13] employed the Burt pyramid to generate fractals for modeling water surfaces. Other methods such as the Fourier transform approach, the fractional Brownian motion, the Shear displacement process and the modified Markov process are well documented [11]. Lovejoy and Mandelbrot [14] propose a method for generating rainfall intensity statistics have fractal properties. This method is called Mandelbrot's fractal sum of pulses (FSP). Pentland [15-16], Medioni, et al. [17], and Peleg, et al. [18] extended the work on fractals by applying them to image texture and computer vision problems. Pentland [15] demostrated that fractal functions (non-differentiable functions) are useful for representing the complex three-dimensional surfaces characterizing natural objects.

The need for such functions results from the fact that the world is full of phenomena that are not orderly or well-behaved. This necessitates a need for a set of functions that are not differentiable and are suitable for mathematical methods dealing with discontinuity and divergence of a phenomenon. Furthermore, the transformation (projection) of a three-dimensional natural scene onto a two-dimensional space (image formation process) does not change the fractal dimension, if the surface is homogeneous. In addition, the fractal model provides a method for image segmentation.

DEVELOPMENT OF THE FRACTAL MODEL AND ITS PARAMETERS

In this research a fractal-based model is developed. The objective is to combine the theory of fractals and image analysis techniques for use in investigating the atmospheric dispersion of hazardous air pollution. Atmospheric structures are three-dimensional and non-isotropic [19] and because of the lack of an adequate geometric framework, these structures cannot easily be classified into a mathematical model. It is hoped that the use of the proposed fractal-based model may be helpful in describing the large number of geometrical structures which are neither ordinary surfaces nor volumes, and contain aspects of dynamic systems showing chaotic behavior [20].

In this new approach, we are not concerned about obtaining the fractal dimension explicitly, however, two parameters are computed, the dynamical horizontal fractal parameter (HFP) and the dynamical vertical fractal parameter (VFP). These two parameters have played an important role in the analysis of our natural data. In general terms, one is always interested in describing how a fractal or near-fractal object can be represented by a measure. A measure is nothing more than a probability allocated to each portion of the object.

We define a random function I(r) as a fractal Brownian function if for all r and variation $\Delta_i r$ (i=1, 2, 3,...) we have:

$$P_r\left[\frac{[I(r+\Delta_i r) - I(r)]}{|\Delta_i r|^{(1 - p_j(v))}} < \theta\right] = F(\theta) \quad , \tag{1}$$

where $F(\theta)$ is a cumulative distribution function, $\Delta_i r$ is the ith element of the vector quantity Δr, and $p_j(v)$ is the fractal parameter which is a function of the variable vector v (vector v may have several variables). The index j stands for either the horizontal or vertical fractal parameter. Physically the parameters $p_j(v)$ describe how the surface of an object varies and indicate how complex a surface is (measure of fragmentation).

Equation (1) can be rewritten as

$$< \Delta I_{\Delta_i r} > | \Delta_i r |^{p_j(v)-1} = < \Delta I_{\Delta_1 r} > \quad , \tag{2}$$

where $< \Delta I_{\Delta_i r} >$ is the mean value of the change in intensity over distance $\Delta_i r$. Using equation (2), we solve for $p_j(v)$ as follows.

$$\log \left[< \Delta I_{\Delta_i r} > | \Delta_i |^{p_j(v)-1} \right] = \log \left[< \Delta I_{\Delta_1 r} > \right] \quad , \tag{3}$$

from which

$$(p_j(v)-1) \log | \Delta_i | = \log < \Delta I_{\Delta_1 r} > - \log < \Delta I_{\Delta_i r} > \; . \tag{4}$$

Furthermore,

$$p_j(v) = 1 + \frac{\log < \Delta I_{\Delta_1 r} > - \log < \Delta I_{\Delta_i r} >}{\log | \Delta_i |} \tag{5}$$

To estimate $p_j(v)$, we need to compute the quantities $< \Delta I_{\Delta_i r} >$ for different spatial separations $\Delta_i r$ (i=1,...). In the two-dimensional case, these spatial separations $\Delta_i r$ may be viewed as concentric pulses representing straight cylinders whose base is a circle. If we consider a reference point x(i,j) in two-dimensional Euclidean space, we then define $\Delta_i r$ as the distance between x(i,j) and its neighbor y(i,j) among points located at the circumference of a disk of radius $\Delta_i r$ and center x(i,j). From the image point of view, the reference point x(i,j) represents the pixel (i,j). The smallest value of the radius $\Delta_i r$ is one unit pixel distance (i=1).

Figure 1 shows the first-order circular neighborhood configuration used to estimate the statistics of the differences of gray levels between pixels representing the fractal behavior of the smoke. The circular neighborhood configuration consists of eight symmetrical pixels, each of which is placed on a circle in the image plane centered at (i,j). For all pairs (i,j), the quadruplet {(i, j+1), (i, j-1), (i+1,j) and (i-1,j)} contains the pixels corresponding to grid corners (rectangular grid) where intensities are known. The remaining four diagonal and off-diagonal locations are not on grid points of a rectangular grid so their intensity values must be interpolated. The interpolation is given by a linear combination of the intensity values of the four nearest grid points multiplied by the normalized inverse Euclidean distance between the interpolated pixel and its neighbors.

Let v = (a,b) be one of the interpolated pixels and $q^{(i)}$, i=1,2,3,4 be the coordinates of the four nearest grid points encircling v. The intensity value at the interpolated pixel is given by

$$f(v) = \left[\sum_{i=1}^{4} d_i \right]^{-1} \left[\sum_{i=1}^{4} d_i f(q^{(i)}) \right] \tag{6}$$

where

$$d_i = \|v - q^{(i)}\|^{-1} \qquad i=1,2,3,4 \tag{7}$$

RESULTS

The concern of the analysis was to extract motion information and to understand how the dynamical fractal parameters are affected by the spatio-temporal distributions of the smoke in the time-varying image sequences. Three different image sequences were used were used to obtain the results. Each sequence had 200 frames whose size was 120 x 350 pixels. The smoke in the three image sequences had speeds of one meter per second, two meters per second, and four meters per second. A window size of 15 x 240 pixels in the turbulent region of the image was chosen to estimate the horizontal fractal parameter. Similarly, a window size of 100 x 15 pixels was chosen to estimate the vertical fractal parameter. Figure 2 shows three frames of a typical sequence at a velocity of 2m/s.

Values of the horizontal and vertical dynamical fractal parameters were computed for 21 frames each (from frame 1 to 200) of each sequence using equation (5). The time interval between frames was 0.33 sec. Figure 3 show the temporal variability of the vertical fractal parameter for the sequences of smoke whose velocity was one, two and four meters per second, respectively. The dispersion of smoke behaves as an oscillatory system which has been reflected in the vertical fractal parameter.

Similarly, figure 4 show the temporal variability of the horizontal fractal parameter for the sequences of smoke whose velocity was one, two and four meters per second, respectively. The fluctuations in the horizontal fractal parameter are larger than for the vertical fractal parameter. The reason for this variation is that over the range of the window the image data is anisotropic.

STATISTICAL INTERPRETATION

In this section, we attempt characterize the statistical attributes of the fractal parameters. In this initial study, we restrict our attention to the case when the fractal parameters vary in terms of time only and the mean value is taken to be zero. Denote ϵ as the deviation from the mean value of the vertical fractal parameter. The standard deviation of ϵ for successive frames will be analyzed to determine its constancy. $\epsilon^{(1)}$ represents the first derivative with respect to time and in general, $\epsilon^{(k)}$ represents the kth derivative.

In the following, we will show the existence of some relationships between ϵ, $\epsilon^{(1)}$,... and the fractal parameters. Assume we have observed the time-samples of ϵ, i.e., $\epsilon(t_1) = \epsilon_1, ..., \epsilon(t_k) = \epsilon_k$. Upon perturbation, the time-sample becomes

$$\epsilon(t_1 + \delta t) = \epsilon_1 + \delta\epsilon_1,\ \epsilon(t_2 + \delta t) = \epsilon_2 + \delta\epsilon_2,\ . . . ,\ \epsilon(t_k + \delta t) = \epsilon_k + \delta\epsilon_k \tag{8}$$

where δt is a small amount of time. Furthermore, if the time t_i $(i = 1,2,...k)$ is taken randomly, we have:

$$< (\epsilon)^2 > = \frac{1}{k}\sum_{i=1}^{k}(\epsilon_i)^2 \tag{9}$$

where "$< >$" denotes the mean value.

Taking the first order terms of each $(\epsilon_i)^2$ yields:

$$<(\epsilon)^2> = \frac{1}{k}\sum_{i=1}^{k}\left(\epsilon_i + \epsilon_i^{(1)}\delta t\right)^2 = <(\epsilon)^2> + 2<\epsilon\epsilon^{(1)}>\delta t \tag{10}$$

To guarantee the near-constancy of the mean $<(\epsilon)^2>$, orthogonality between ϵ and $\epsilon^{(1)}$ must hold, i.e.,

$$<\epsilon\epsilon^{(1)}> \equiv 0 \tag{11}$$

Differentiating (11) yields,

$$<\epsilon\epsilon^{(2)}> + <(\epsilon^{(1)})^2> = 0 \tag{12}$$

Applying the definition of the correlation coefficient, we get

$$\rho = -\frac{<(\epsilon^{(1)})^2>}{<(\epsilon)^2>^{1/2}<(\epsilon^{(2)})^2>^{1/2}} \tag{13}$$

Assuming that the standard deviation of $\epsilon^{(1)}$ is nearly constant allows us to use the same logic as for the standard deviation of ϵ. In general, this logic may be applied to all higher derivatives and we end up with:

$$<\epsilon\epsilon^{(2k)}> = (-1)^k <(\epsilon^{(k)})^2> \tag{14}$$

and

$$<\epsilon\epsilon^{(2k+1)}> = 0 \tag{15}$$

These concepts can now be applied to the time-sample of ϵ, $(\epsilon(t_i) = \epsilon_i$, i=1,2,...,k). If τ represents a small amount of time, the correlation coefficient is

$$\rho_\tau = \frac{<\epsilon(t)\epsilon(t+\tau)>}{<(\epsilon(t))^2>^{1/2}<(\epsilon(t+\tau))^2>^{1/2}} \tag{16}$$

If we assume near-constancy for the standard deviation of ϵ, (16) becomes

$$\rho_\tau = \frac{<\epsilon(t)\epsilon(t+\tau)>}{<(\epsilon(t))^2>} \tag{17}$$

If we expand $\epsilon(t+\tau)$ in powers of τ, (17) becomes:

$$\rho_\tau = 1 - \frac{\tau^2}{2!}\frac{<(\epsilon^{(1)})^2>}{<(\epsilon)^2>} - \cdots + \frac{(-1)^{(k)}\tau^{2k}}{(2k)!}\frac{<(\epsilon^{(k)})^2>}{<(\epsilon)^2>} \tag{18}$$

To give a flavor for equation (18), consider $\epsilon = \sin(t + \xi)$, where $\xi \in \mathbb{R}$. Since,

$$<(\epsilon)^2> = <(\epsilon^{(1)})^2> = \cdots = <(\epsilon^{(k)})^2> = 0.5 \tag{19}$$

equation (18) becomes,

$$\rho_\tau = \sum_{k=0}^{\infty} (-1)^{(k))} \frac{\tau^{2k}}{(2k)!} = \cos\tau \qquad (20)$$

This supports our earlier observation of the temporal variability in the vertical fractal parameter.

SUMMARY

In this paper, a new model based on fractal sets is used to analyze video images of smoke data. We have chosen to study two parameters, the horizontal fractal parameter and the vertical fractal parameter. Their values have been computed and the relevant properties of the turbulent motion and fluctuations of the pollutant at some points of interest are inferred. Our first concern is the smoke, but the approach can be used for analyzing other phenomenal objects.

In our analysis, the input data consists of three image sequences taken at three different velocities in a wind tunnel. Analysis and testing of the technique have produced a series of graphical relations between the relevant features of smoke. These graphical relations tell us information about which smoke features can be determined from the image sequences. Furthermore, we attempt to explain and interpret the obtained results by means of a statistical theory.

REFERENCES

[1] Robins, A.G. and I.P. Castro, "A Wind Tunnel Investigation of Plume Dispersion in the Vicinity of a Surface Cube: Flow and Concentration Field," *Atmosph. Environ.,* 11, pp. 291-311, 1977.

[2] Frost, W., J. Bitte and C. Shieh, "Analysis of Neutrally Stable Atmospheric Flow over a Two-dimensional Forward-facing Step," *A.I.A.A.J.,* 18:1, pp. 32-38, 1980.

[3] Hunt, J.C.R., R.E. Briteer and J.S. Puttock, "Mathematical Models of Dispersion of Air Pollution Around Buildings and Hills," in Harris (ed), *Mathematical Modeling of Turbulent Diffusion in the Environment,* Academic Press, pp. 145-200, 1979.

[4] Millan, M.M. and R.M. Hoff, "Dispersive Correlation Spectroscopy: A Study of Mask Optimization Procedures," *Applied Optics,* 16, pp. 1609-1618, 1977.

[5] Millan, M.M. and R.M. Hoff, "Remote Sensing of Air Pollutants by Correlation Spectroscopy Instrumental Response Characteristics," *Atmosph. Environ.,* 12, pp. 853-864, 1978.

[6] Collis, R.T.H. and P.B. Russel, "Lidar Measurement of Particles and Gases by Elastic Backscattering and Differential Absorption," in *Laser Monitoring of the Atmosphere,* (ed. E.D. Hinkley), 14, pp. 71-145, 1976.

[7] Byer, R.L. and M. Garbuny, "Pollutant Detection by Absorption Using Mie Scattering and Topographic Targets as Retroflectors," *Applied Optics,* 12, pp. 1496-1505, 1973.

[8] Bowre, N.E., R.J. Londergan, D.R. Murray and H.S. Borenstein, EPRI Report EA-3074, Electric Power Research Institute, Palo Alto, California, 1983.

[9] A.R. Mahoney, L.G. Mcallister and J.R. Pillard, "The Remote Sensing of Wind Velocity in the Lower Troposphere Using Acoustic Sounder," *Boundary-Layer Meteorol,* 4, pp. 155-167, 1973.

[10] Mandelbrot, Benoit B., *Les Objets Fractals: Forme, Hazard et Dimension,* Paris: Flammarion, 1975.

[11] Mandelbrot, Benoit B., *The Fractal Geometry of Nature,* San Francisco, CA, 1983.

[12] Willson, Stephen J., "Cellular Automata Can Generate Fractals," *Discrete Applied Mathematics,* 8, pp. 91-99, 1984.

[13] Odgen, J., Generation of Fractals Using the Burt Pyramid," Optical Society of America Annual Meeting Washington, D.C., Oct. 1985.

[14] Lovejoy, S. and Mandelbrot, B.B., "Fractal properties of Rain, and A Fractal Model," *Tellus,* 37 A, pp. 209-232, 1985.

[15] Pentland, A.P. "Fractal-based Description of Natural Scenes," *PAMI-6,* pp. 661-674, 1984.

[16] Pentland, A.P. "Shading into Texture," *Artificial Intelligence,* 29, pp. 147-170, 1986.

[17] Medioni, Gerard and Y. Yasumimoto, "A Note on Using the Fractal Dimension for Segmentation," IEEE Computer Vision Workshop, Annapolis, MD, 1984.

[18] Peleg

[19] Lovejoy S. and D. Schertzer, "La geómetrie fractale des nuages et des regions de pluie et les simulation Aléatoires," *La Houille Blanche,* No. 5/6, 1983.

[20] M.W. Hirsch, "The Chaos of Dynamical Systems," in *Chaos, Fractals, and Dynamics* (ed. P. Fischer, W.R. Smitch) pp. 189-196, 1985.

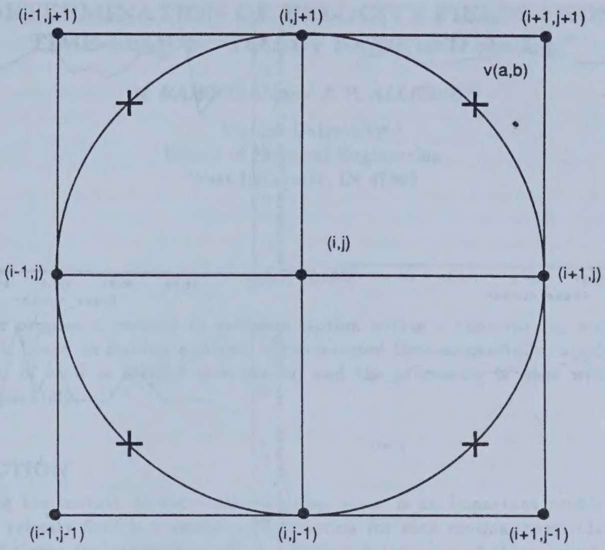

(i-1,j+1)
(i,j+1)
(i+1,j+1)
v(a,b)
(i,j)
(i-1,j)
(i+1,j)
(i-1,j-1)
(i,j-1)
(i+1,j-1)

Fig. 1. First-order circular neighborhood configuration ($\Delta_1 x$)

● real pixel

⊕ interpolated pixel

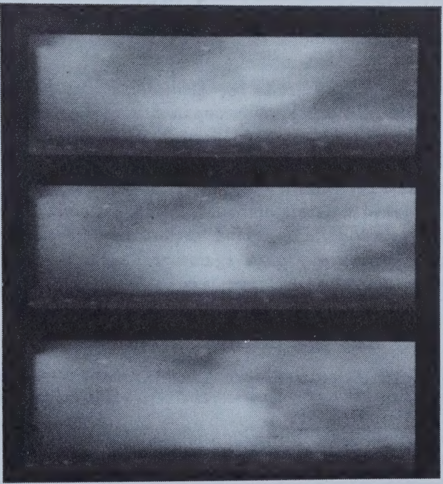
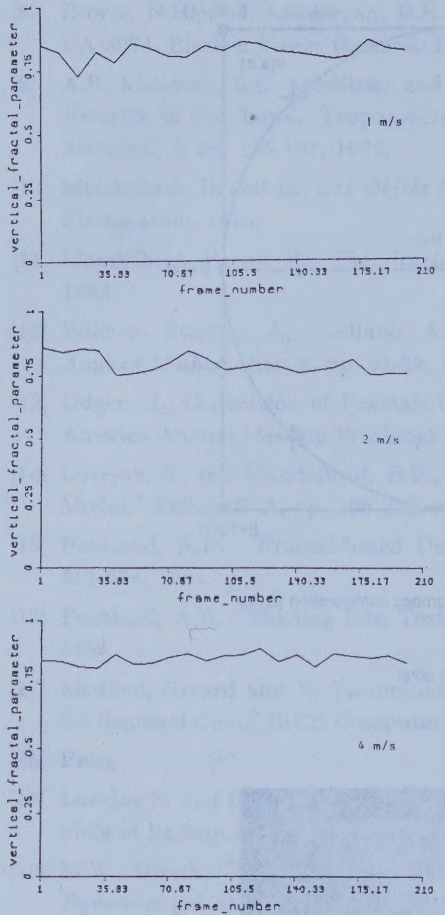
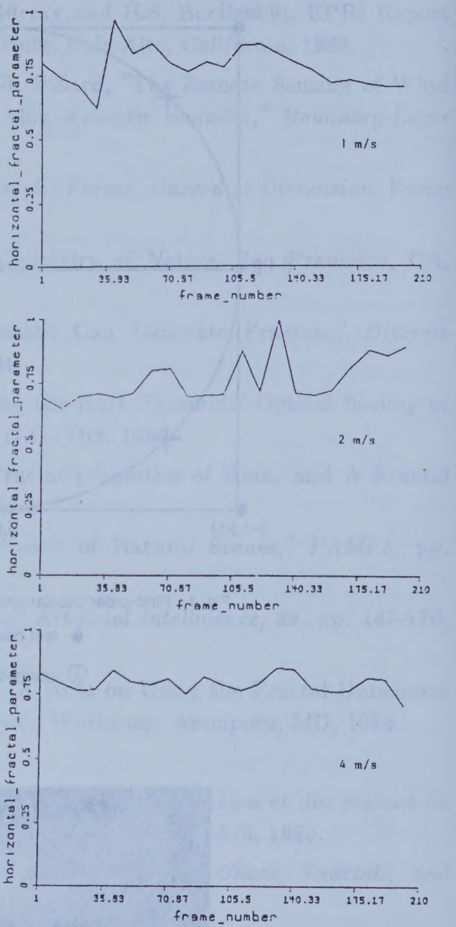


Fig. 2 Video Image Data



Vertical fractal parameter versus frame number (time) case: freestream wind speed of 1, 2, 4 m/s

Fig. 3



Horizontal fractal parameter versus frame number (time) case freestream wind speed of 1, 2, 4 m/s

Fig. 4

DETERMINATION OF VELOCITY FIELDS FROM TIME-SEQUENTIALLY SAMPLED DATA *

M. RAHGOZAR and J. P. ALLEBACH

Purdue University
School of Electical Engineering
West Lafayette, IN 47907

ABSTRACT

We propose a method to estimate motion within a time-varying scene. This method is based on Fourier analysis of bit-reversed time-sequentially sampled data. The field of view is divided into blocks, and the processing is done within each block separately.

1. INTRODUCTION

Determining the motion within a time-varying scene is an important problem in image processing [1]. The velocity field is a useful representation for such motion, particularly in a complex scene. Most techniques for generating velocity fields fall into one of three broad catagories, each one of which has its own shortcomings: 1. Gradient intensity methods are highly iterative algorithms based on differentials in image intensities which relate local temporal changes to local spatial changes in an image. Since they are local methods, they are susceptible to noise and distortions, and are computationally intensive [2]. 2. Matching techniques seek to identify the movement of certain highly localized features such as edges and corners from one frame to the next. Inherently associated with these techniques is the correspondence problem between identifiable features in consecutive frames. Again, because of localization, this method is strongly influenced by noise or distortion. 3. In contrast, Fourier techniques incorporate data from all the points within a neighborhood over which the transform is computed. Analyzing the disparity in the local Fourier spectra of two consecutive frames will yield information about motion in the field of view (FOV). For any of these techniques to be successful, there must be relatively little change between consecutive frames.

The approaches to motion estimation discussed above have been based on data taken frame-instantaneously; i.e., a full frame of data is acquired from the FOV during a time interval that is short compared to the interval between frames of data. In practice, however, most sampling mechanisms operate in a time-sequential fashion. In this mode, samples are taken from the FOV, one at a time, according to some specific ordering which is repeated from frame to frame. Scanning mechanisms that may operate time-sequentially include electron beam deflection, random addressing of sensor arrays, electromechanical and/or holographic spot deflection, and phased element array beamformers. The most familiar orderings used are lexicographic (LEX) and 2:1 line-interlaced patterns which are used in video systems. Other patterns include bit-reversed (BRV), congruential, and random orderings [3,4].

---

* This work was supported by the Department of Energy under Contract DE-AC03-87SF16732, and the National Science Foundation under Grant ECS-8419997. M. Rahgozar was a David Ross Fellow while this work was performed.

We have shown previously that time-sequential sampling according to novel orderings can provide much better resolution of spatiotemporal frequency components than frame-instantaneous or lexicographic sampling at the same data rate [5]. These novel patterns disperse consecutive samples over the entire FOV. This work was entirely sampling-theoretic in nature. We also investigated the problem of determining the motion of a small maneuvering target with data acquired time-sequentially [6]. We found that surprisingly accurate estimation of target motion was possible from a single frame of data even when there is significant motion during the frame interval.

In this paper, we consider the problem of generating a velocity field with time-sequentially acquired data. Based on our earlier results, we would expect to be able to obtain accurate velocity fields even when there is significant motion during the interval in which a single frame of data is acquired. However, to achieve this performance, we must sample with a pattern for which the samples in any small neighborhood of the FOV will be temporally distributed over the entire frame interval. This makes it difficult to apply either gradient intensity or matching methods for determining the velocity field. On the other hand, this poses no problem with the Fourier-based approach which we consider here. Use of Fourier techniques to generate velocity fields with frame-instantaneous data has previously been considered by Groves, Rajala, and Snyder [7]. In their approach, the FOV is partitioned into blocks; and each block is processed independently to generate an estimate of the velocity vector within that block. To process time-sequentially acquired data in blocks, it is important that the structure of the sampling pattern be the same within each block. Fortunately, the BRV sampling pattern is a dispersed pattern that has this property.

The remainder of the paper is organized as follows. In Sec. 2, we describe our model for the motion within each block. In Sec. 3, we describe the BRV sampling pattern, and derive its structure within each block. In Sec. 4, we present our algorithm for determining the velocity vector within each block. In Sec. 5, we present our preliminary results with this algorithm. Finally, in Sec. 6, we make our conclusions, and discuss some additional topics.

2. MOTION MODEL

So far, we have discussed only signals that depend on two spatial dimensions and time. However, one might also be interested in estimating motion in three spatial dimensions. In fact, the concepts to be presented here may easily be extended to any function that is evolving with time and N independent variables. In this paper, we shall work within this more general setting. However, we will still refer to the vector $\mathbf{x}$ as the spatial coordinate.

The temporal evolution in a time-varying scene may be quite complex. But, during a relatively short interval, the motion in any small part of the FOV may be approximated as simple translation with constant speed. Accordingly, we divide the FOV into blocks and process each block independently to determine the parameters of the motion within that block.

Let $o(\mathbf{x})$ represent the spatial variation of the scene irradiance within any given block. Under the assumption of translational motion with constant speed, the irradiance at the focal plane will be given by

$$g(\mathbf{x},t) = o(\mathbf{x}-\mathbf{V}t), \tag{1}$$

where $\mathbf{V}$ is the velocity vector. In the spatiotemporal frequency domain, we find that

$$G(\mathbf{u},f) = O(\mathbf{u})\delta(f+\mathbf{V}\cdot\mathbf{u}). \tag{2}$$

Here, $\mathbf{u}$ is the N-D spatial frequency vector, and $O(\mathbf{u})$ is the spectrum of the component $o(\mathbf{x})$ of spatial variation within the block. From (2), we see that for translational motion with constant speed, the spectral energy in the time-varying scene irradiance is confined to the hyperplane $f+\mathbf{V}\cdot\mathbf{u}=0$. The distributon of the spectral energy on this hyperplane is determined by the spectrum $O(\mathbf{u})$ of the spatial variation in the scene. The orientation of the hyperplane in (N+1)-D frequency space is determined by the vector $\mathbf{V}$. We propose a scheme to estimate this vector for each block.

3. BLOCK PROCESSING OF DATA ACQUIRED TIME-SEQUENTIALLY IN BIT REVERSED ORDER

We begin by establishing some notation that will simplify the mathematics. As we have so far, we denote N-tuples of variables by a boldface version of the variable name. We shall apply the usual operations for scalars to these vectors with understanding that the result will be a vector obtained by carrying out the operation element by element. For example, if $\mathbf{x} = (x_1, \cdots, x_N)$ and $\mathbf{y} = (y_1, \cdots, y_N)$, then $\mathbf{xy} = (x_1y_1, \cdots, x_Ny_N)$, and

$$\sum_{\mathbf{a}=\mathbf{0}}^{\mathbf{M}-\mathbf{1}} = \sum_{a_1=0}^{M_1-1} \cdots \sum_{a_N=0}^{M_N-1} .$$

We also define the usual inner product as $\mathbf{x}\cdot\mathbf{y} = x_1y_1 + \cdots + x_Ny_N$, and an additional operator $[\mathbf{x}]_\bullet = x_1 \cdots x_N$.

Now consider a process $g(\mathbf{x},t)$ which depends on N independent variables $\mathbf{x} = (x_1, \cdots, x_N)$ and time t. We sample each of the variables x_j at M_j points uniformly spaced in the interval $[0,A_j]$, $j=1, \cdots, N$; and we take one sample every T seconds. At time kT, we sample at $\mathbf{x} = (\boldsymbol{\alpha}_k\mathbf{X})$ where $\mathbf{X} = \mathbf{A}/\mathbf{M}$. The vector index sequence $\boldsymbol{\alpha}_k = (\alpha_{1k}, \cdots, \alpha_{Nk})$ defines the order in which the points are sampled. We shall assume that $\boldsymbol{\alpha}_k$ is a permutation of all N-tuples of integers $\mathbf{a}$, $\mathbf{0} \leq \mathbf{a} \leq \mathbf{M}-\mathbf{1}$ for $k = 0, \ldots, [\mathbf{M}]_\bullet-1$. Outside this range, we assume that the sequence repeats periodically; so $\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_{k+b[\mathbf{M}]_\bullet}$ for integer b. We obtain an entire frame of samples in $B=[\mathbf{M}]_\bullet T$ s. The time-sequentially sampled signal may be represented by

$$\tilde{g}(\mathbf{x},t) = [\mathbf{X}]_\bullet B \sum_k g(\boldsymbol{\alpha}_k\mathbf{X},kT)\delta(\mathbf{x}\_\boldsymbol{\alpha}_k\mathbf{X},t-kT). \tag{3}$$

Here $[\mathbf{X}]_\bullet B$ is the hypervolume of each cell; and $\delta(\mathbf{x},t)$ is the (N+1)-dimensional Dirac delta function.

Of all the dispersed sampling patterns that we have studied to date, the BRV pattern appears most suitable for block processing. This is because within each block, the pattern has the same structure as over the entire FOV. Fig. 1 shows an example of this pattern for the case of N=2 spatial dimensions with $M_1 = M_2 = 16$. It has the following structure: Every 4-th sample falls in the same quadrant of the FOV. If we divide any quadrant into subquadrants, we find that every 4-th sample in that quadrant falls in the same subquadrant. We may repeat this process by further dividing subquadrants until each subquadrant contains only one sample point. This defines the structure of the entire sampling pattern. In general N-D space, the BRV pattern may be represented by the relations

$$\alpha_{jk} = \sum_{l=0}^{L-1} k_{lN+j-1}\, 2^{L-l-1}, \tag{4}$$

$$k = \sum_{l=0}^{NL-1} k_l\, 2^l, \tag{5}$$

where $M_j = 2^L$ for $j=1, \cdots, N$ and the sample at time $t=kT$ is taken at $\boldsymbol{\alpha}_k\mathbf{X} = (\alpha_{1k}X_1, \cdots, \alpha_{Nk}X_N)$. Now suppose we divide the FOV into blocks of size $M^0 \times \cdots \times M^0$, where $M^0 = 2^{L^0}$. In order to study the structure of the sampling pattern relative to these blocks, we will make the following distinctions. Refrences to the entire FOV will be made with no superscripts, those to the blocks as a whole with superscript 1, and those to points within the blocks with superscript 0. Thus, α_{jk}^1, $j=1, \cdots, N$ designates the block to be sampled at time $t=kT$, and α_{jk}^0, $j=1, \cdots, N$ gives the coordinates of that sample within the block. For any sampling pattern,

$$\alpha_{jk} = \alpha_{jk}^1 M^0 + \alpha_{jk}^0\,, \qquad j=1, \cdots, N. \tag{6}$$

For the BRV pattern,

$$\alpha_{jk}^0 = \sum_{l=0}^{L^0-1} k_{N(l+L^1)+j-1}\, 2^{(L^0-1)-l}, \tag{7}$$

and

$$\alpha_{jk}^1 = \sum_{l=0}^{L^1-1} k_{lN+j-1}\, 2^{(L^1-1)-l}, \tag{8}$$

where $L^1 = L - L^0$. From (7), we can see that the sampling pattern within each block is a bit reversed mapping of the upper L^0 bits of the time index given by (5) to the spatial coordinates within block.

This pattern is of the same form as that for the entire FOV given by (4). Similarly, (8) shows that the individual blocks are accessed in bit reversed order according to the lower L^1 bits of the time index.

Since the sampling pattern is the same within each block, we may do the same processing within each block to estimate the velocity vector. In addition, since the sampling pattern within the block is BRV, we still obtain superior resolution of spatiotemporal frequency components that we have previously demonstrated with respect to the entire FOV [8].

Having determined the structure of the sampling pattern within each block, we now restrict our attention in the remainder of this paper to a single block. We treat that block as a FOV containing a small patch of the time-varying scene and carry out the analysis using general notation, keeping in mind that the same processing is to be done on all the blocks. As mentioned in Sec. 2, we would expect under the assumption of constant speed translation to find all the spectral energy in the time-varying patch confined to a hyperplane in the frequency domain. This relation is expressed in (2) in terms of the continuous Fourier transform (CFT) of the continuous-parameter, time-varying scene $g(\mathbf{x},t)$. To use this result for motion estimation, we need to relate it to the time-sequentially sampled data $\tilde{g}_k = g(\alpha_k\mathbf{X},kT)$ for $k = 0, \cdots, [\mathbf{M}]_\bullet - 1$, on which the estimate will be based. Consider the (N+1)-D CFT $\tilde{G}(\mathbf{u},f)$ of the scanned signal represented by a sample-modulated impulse train as in (3). When evaluated at points $\mathbf{u} = \frac{\mathbf{m}}{\mathbf{MX}}$ and $f = \frac{p}{[\mathbf{M}]_\bullet T}$, we obtain

$$\tilde{G}_{\mathbf{m}p} = \tilde{G}\left(\frac{\mathbf{m}}{\mathbf{MX}}, \frac{p}{[\mathbf{M}]_\bullet T}\right) \tag{9a}$$

$$= \frac{1}{[\mathbf{M}]_\bullet} \sum_{k=0}^{[\mathbf{M}]_\bullet - 1} g_k \, e^{-i2\pi\left(\alpha_k \cdot \frac{\mathbf{m}}{\mathbf{M}} + \frac{pk}{[\mathbf{M}]_\bullet}\right)}, \tag{9b}$$

which is the (N+1)-D discrete Fourier transform (DFT) of the data. We have shown that $\tilde{G}(\mathbf{u},f)$ is related to the spectrum $G(\mathbf{u},f)$ of the continuous-parameter time-varying scene by [8]

$$\tilde{G}(\mathbf{u},f) = \sum_{\mathbf{j}}\sum_{l} Q_{\mathbf{j}l} \, G\left(\mathbf{u} - \frac{\mathbf{j}}{\mathbf{A}}, f - \frac{l}{B}\right), \tag{10}$$

where

$$Q_{\mathbf{j}l} = \frac{1}{[\mathbf{M}]_\bullet} \sum_{k=0}^{[\mathbf{M}]_\bullet - 1} e^{-i2\pi\left(\alpha_k \cdot \frac{\mathbf{j}}{\mathbf{M}} + \frac{lk}{[\mathbf{M}]_\bullet}\right)}. \tag{11}$$

Evaluating (10) on the reciprocal sampling lattice indicated by (9a), we obtain

$$\tilde{G}_{\mathbf{m}p} = \sum_{\mathbf{j}}\sum_{l} Q_{\mathbf{j}l} \, G\left(\frac{\mathbf{m}-\mathbf{j}}{\mathbf{A}}, \frac{p-l}{B}\right). \tag{12}$$

This is nothing more than circular convolution of the (N+1)-D DFT of the coefficients $Q_{\mathbf{j}l}$ of the time-sequential sampling pattern and the (N+1)-D DFT of the time-varying patch sampled on a non-time-sequential lattice with interval $\mathbf{X} \times T$.

Recalling that the motion estimation algorithm is to find the orientation of the hyperplane $f + \mathbf{u} \cdot \mathbf{V} = 0$ in the frequency domain, we need to be concerned with the effects of the aliasing, indicated by (12). Here, we will only point out that for the BRV sampling pattern, the coefficients $Q_{\mathbf{j}l}$ are relatively large in the vicinity of the $(\mathbf{j},l) = (\mathbf{0},0)$ baseband term. This fact could degrade our estimate for certain ranges of patch velocity $\mathbf{V}$.

Since the DFT of one frame of the data only provides samples from the CFT $G(\mathbf{u},f)$ of the time-varying scene patch at interval $1/B$ along the temporal frequency axis, we cannot expect to resolve motion corresponding to temporal frequencies $<1/B$. To extend the resolution to lower frequencies, we can take one of the following two approaches. We can process K frames of data simultaneously. In this case, the (N+1)-D DFT of the data is given by

$$\tilde{G}_{mp}=\frac{1}{K[M]_\bullet}\sum_{k=0}^{K[M]_\bullet-1} g_k\, e^{-i2\pi(\alpha_k\cdot\frac{m}{M}+\frac{pk}{K[M]_\bullet})}, \tag{13}$$

which, neglecting aliasing, provides samples from the CFT of the time-varying patch at the interval 1/KB along the temporal frequency axis; so

$$\tilde{G}_{mp}=G(\frac{m}{A},\frac{p}{KB}).$$

Alternatively, we can interpolate the spectral components for lower frequencies. This can be done by appending the approperiate number of zeros to the data before taking the 1-D DFT of the sequence $g_k e^{-i2\pi(\alpha_k\cdot\frac{m}{M})}$ as indicated by (9b). Appending $(K-1)[M]_\bullet$ zeros would interpolate the spectrum down to intervals of 1/KB. We choose to take this last approach in order to investigate the capability of our motion estimation algorithm using only one frame of data. However, we will carry out our derivation for K frames in order to show its ability to handle multiple frames of data.

We conclude this section with a final note on using the DFT. The assumption underying all DFT calculations is that the original continuous-parameter signal is either zero outside both temporal and spatial windows or repeats periodically. To reflect this fact, we incorporate the finite duration of the measurement windows in our expression for the time-varying patch, thus obtaining

$$g(x,t) = o(x-Vt)\,\mathrm{rect}(\frac{t}{KB})\,\mathrm{rect}(\frac{x}{A}), \tag{14}$$

where rect(x) = 1, if $|x| \leq 1/2$ and 0, else. Depending on the patch's velocity, the implied window in spatial domain would act as a temporal window which causes the time-varying signal to be zero after a certain elapsed time. As a result, the hyperplane $f+V\cdot u=0$ will be blurred along the f-axis by a sinc function corresponding to the smaller of the two windows in the time domain where, $\mathrm{sinc}(x) = \sin(\pi x)/\pi x$. In cases where the patch is mostly visible in the block during the entire observation period,

$$G(u,f) = KB\, O(u)\, \mathrm{sinc}(\, KB\,(f+V\cdot u)). \tag{15}$$

4. MOTION ESTIMATION

To find the velocity vector inside each block, we need to estimate the orientation of the $f+u\cdot V=0$ hyperplane within that block. The actual data in the (u,f) domain may not, however, be confined to this plane. This could be due to one or more of a number of reasons: 1) time variation within the block which is not exactly constant speed translation; 2) corruption of the data by noise; 3) the effect of aliasing; and 4) the finite temporal duration of the data. The last two sources of nonideality were discussed in the previous section. For each u, we may visualize these sources of nonideality as smearing the spectral energy of the time-varying patch along the f-axis. Our first step toward motion estimation will then be to pin down a particular $\hat{f}$ associated with each u. In view of the discussions in the previous section, we choose $\hat{f}$ to be the peak of the sinc function along the f-axis. We then take as our estimate of velocity the value $\hat{V}$ which gives a weighted least squares fit between $\hat{f}$ and the plane $f = -\hat{V}\cdot u$.

To make this more precise, we first observe that the (N+1)-D DFT of K frames of the sampled data may be expressed as

$$\tilde{G}_{mp}= KB\, O(\frac{m}{MX})\, \mathrm{sinc}(p+\overline{V}\cdot m), \tag{16}$$

where $\overline{V} = KBV/A$ is the normalized velocity vector expressed in units of FOV's / measurement interval, and m and p are discrete variables as before. In what follows, we work with these quantities. In writing (16), we have neglected aliasing.

Our estimate of the velocity is updated continuously by the motion estimation algorithm, one cycle of which is as follows: First for each m, we search along the f-axis in a prescribed interval for a frequency $\hat{p}_m$ at which the spectral data is a maximum compared to the other points in that interval. Let $\hat{O}_m$ be the value of the spectral component at this point. We justify this choice by stating that although the spectral data is smeared along the f-axis, the peak of the sinc function is a good indicator of where the true p_m should be. Next, we update our estimate of $\overline{V}$, by seeking that value $\hat{V}$ which minimizes

$$\Phi = \sum_{m=0}^{\overline{M}} |\hat{O}_m|^2 \left[\hat{p}_m + (\overline{V}\cdot m)\right]^2 . \tag{17}$$

Here, the summation is over all values of $\mathbf{m}$ visited so far. We indicate this fact by putting a bar over $\mathbf{M}$. With this notation, we emphasize the point that the N-D spatial frequency space need not be searched in the usual lexicographic order. In fact, in our implementation of this algorithm we have traversed the space radially outward starting from the $\mathbf{m} = \mathbf{0}$ point. Finally, the updated velocity estimate predicts a pivot point p_o around which we seek the local maximum for the next value of $\mathbf{m}$. We stop updating the velocity estimate when the maximum $\hat{O}_m$ is below some threshold for some number of $\mathbf{m}$'s. The threshold and the number of points used to stop the recursion are heuristics which are selected to avoid aliasing as much as possible. We have chosen (17) as our error measure instead of one that penalizes against deviations of the distance to the plane, since we expect that all the corruption will only be present along the f-axis.

Differentiating (17) with respect to $\overline{V}_j$, $j=1, \cdots, N$, we find that the least squares velocity estimate is given by the solution to the following set of equations which are linear in the unknown $\hat{V}$

$$\sum_{m=0}^{\overline{M}} |\hat{O}_m|^2 \left[\hat{p}_m + \hat{V}\cdot m\right] m_j = 0, \qquad j=1, \cdots, N. \tag{18}$$

For N=2 spatial dimensions, the solution to these equations is given by

$$\hat{V}_1 = \frac{S_1T_2 - S_2T_{12}}{T_{12}^2 - T_1T_2}, \qquad \hat{V}_2 = \frac{S_2T_1 - S_1T_{12}}{T_{12}^2 - T_1T_2}, \tag{19}$$

where

$$S_1 = \sum_{m=0}^{\overline{M}} m_1 |\hat{O}_m|^2 \hat{p}_m, \qquad S_2 = \sum_{m=0}^{\overline{M}} m_2 |\hat{O}_m|^2 \hat{p}_m, \tag{20}$$

$$T_1 = \sum_{m=0}^{\overline{M}} m_1^2 |\hat{O}_m|^2, \qquad T_2 = \sum_{m=0}^{\overline{M}} m_2^2 |\hat{O}_m|^2, \qquad T_{12} = \sum_{m=0}^{\overline{M}} m_1 m_2 |\hat{O}_m|^2. \tag{21}$$

5. RESULTS AND DISCUSSION

Our early experimentation with the motion estimation algorithm has been with K = 1 frame of synthetic imagery. We have chosen a Gaussian object $o(x) = 255\exp[-\pi|x|^2/\rho^2]$ undergoing constant speed translational motion. The FOV is 64×64 and blocks are each 16×16. For an object with $\rho = 4$, we have experimented with speeds in the range $2\sqrt{2}, \cdots, 8\sqrt{2}$ pixels / frame. Velocity field representations for this case are shown in Fig. 2. The estimates associated with this set of velocities are very accurate. With these choices, we are essentially keeping the bright part of the object inside a block during the time it takes to collect one frame of data. As expected, the validity of the estimates decreases for faster speeds. This is mainly due to aliasing. As the speed increases, the hyperplane tilts more toward the f-axis; and so it will pick up more contributions from other spectral replications. The velocity estimates deteriorate accordingly.

Theoretically, we should expect to do better with bigger block sizes. However, in general, the constant speed translation assumption would be violated for a bigger block size. Fig. 3 illustrates velocity fields for a much larger Gaussian object; ρ=16. Estimates for this case are not as good as before. For this object, there is an additional source of error due to the use of the DFT. Since the DFT assumes periodicity, to get a valid representation of the spectrum, the signal needs to be continuous across the boundary. But with an object which is bigger than a block, this source of nonideality is unavoidable. This discontinuity of gray values introduces extraneous high frequency components in the spectrum of the time-varying signal, which in effect deteriorate our estimate.

6. CONCLUSION

We have presented a method for estimating the velocity field in a time-varying scene based on time-sequentially acquired data. The method is based on block processing of the data within the field of view (FOV). As a result of our previous investigation of various time-sequential sampling patterns, we have chosen the bit-reversed (BRV) sampling pattern for its capability to resolve higher temporal frequencies. We have also shown that with the BRV pattern, we have the added advantage that the sampling pattern inside each block of the FOV follows the same ordering as that in the entire FOV. This enables us to process each block identically. Inside each block, we have shown that the spectral energy of a moving patch is confined to the hyperplane $f+u \cdot V=0$, assuming the motion is simple enough to be approximated as constant velocity translation. We have proposed a scheme to estimate the orientation of this hyperplane in $(N+1)$-D frequency domain. We have first localized a temporal frequency for each spatial frequency u, individually, and then have estimated the orientation of the hyperplane as that of one which best fits these localized frequencies. Neglecting aliasing, we have presented a set of linear equations that the components of the velocity estimate must satisfy. These estimates are continuously updated.

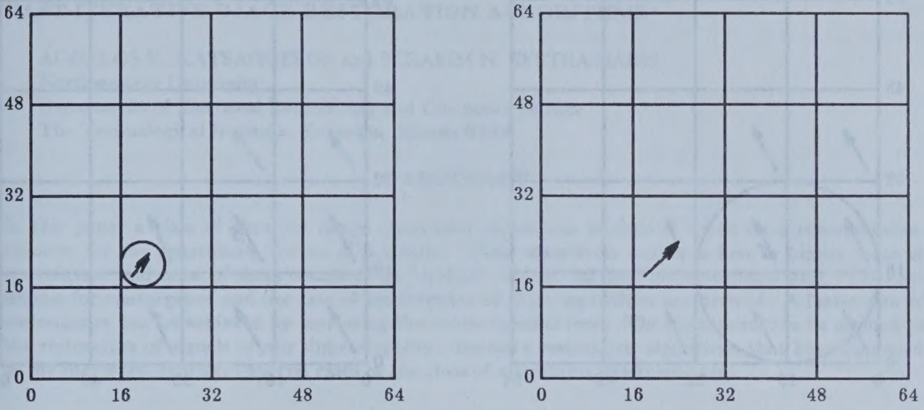
For small enough velocities, such that aliasing is not a factor, we have obtained very accurate estimates of velocities, using only one frame of data. Our future work in this area will address such issues as assessing the validity of the constant speed translational model based on the acquired data itself, the effect of multiple frames, and a detailed analysis of aliasing. We also plan to investigate whether there is a class of congruential patterns that have the type of good block structure possessed by BRV patterns. In terms of aliasing, we believe that such patterns are superior to the BRV patterns.

References

1. T. S. Huang, *Image Sequence Analysis,* Springer-Verlag, New York, 1981.
2. B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.
3. J. P. Allebach, "Analysis of Sampling-Pattern Dependence in Time-Sequential Sampling of Spatiotemporal Signals," *J. Opt. Soc. Am.*, vol. 71, pp. 99-105, January 1981.
4. J. P. Allebach, "Design of Antialiasing Patterns for Time-Sequential Sampling of Spatiotemporal Signals," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-32, pp. 137-144, February 1984.
5. J. P. Allebach, "Effect of Sample Displacement in Time-Sequential Sampling of Spatiotemporal Signals," *J. Opt. Soc. Am.*, vol. 73, pp. 466-472, Apr 1983.
6. J. B. Koskol and J. P. Allebach, "Estimating the Motion of a Maneuvering Target with Time-Sequentially Sampled Imagery," *J. Opt. Soc. Am.*, vol. JOSA-A 3, pp. 1504-1511, 1986.
7. M. H. Groves, S. A. Rajala, W. E. Snyder, "Displacement Field Calculation by the Motion Detection Transform with Application in FLIR Imagery," *Proceedings of the 1984 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1984.
8. J. P. Allebach, "Time-Sequential Sampling in N-Dimensional Euclidean space," *Proceedings of the 21st Annual Allerton Conference on Communications, Control, and Computing*, pp. 110-119, Monticello, IL., Oct. 5-7 1983.

0	64	16	80	4	68	20	84	1	65	17	81	5	69	21	85
128	192	144	208	132	196	148	212	129	193	145	209	133	197	149	213
32	96	48	112	36	100	52	116	33	97	49	113	37	101	53	117
160	224	176	240	164	228	180	244	161	225	177	241	165	229	181	245
8	72	24	88	12	76	28	92	9	73	25	89	13	77	29	93
136	200	152	216	140	204	156	220	137	201	153	217	141	205	157	221
40	104	56	120	44	108	60	124	41	105	57	121	45	109	61	125
168	232	184	248	172	236	188	252	169	233	185	249	173	237	189	253
2	66	18	82	6	70	22	86	3	67	19	83	7	71	23	87
130	194	146	210	134	198	150	214	131	195	147	211	135	199	151	215
34	98	50	114	38	102	54	118	35	99	51	115	39	103	55	119
162	226	178	242	166	230	182	246	163	227	179	243	167	231	183	247
10	74	26	90	14	78	30	94	11	75	27	91	15	79	31	95
138	202	154	218	142	206	158	222	139	203	155	219	143	207	159	223
42	106	58	122	46	110	62	126	43	107	59	123	47	111	63	127
170	234	186	250	174	238	190	254	171	235	187	251	175	239	191	255

Fig. 1. Bit reversed sampling pattern for N = 2 spatial dimensions and a 16×16 field of view. Each number indicates the order in which that spatial point would be sampled.

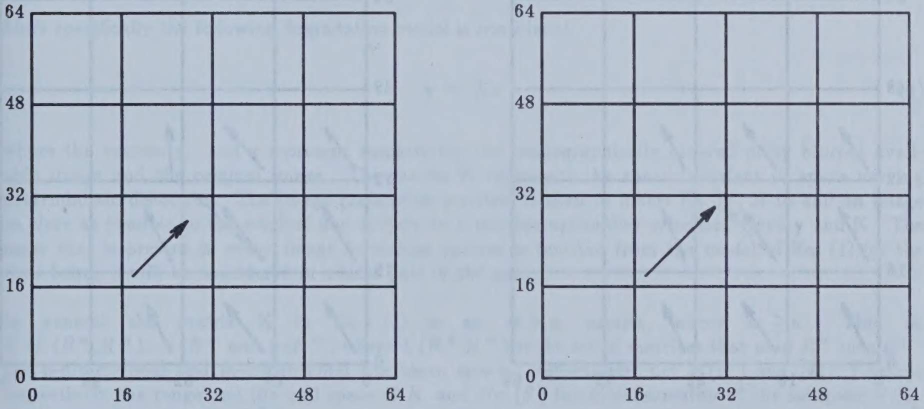


a

speed = $2\sqrt{2}$ pixels / frame

b

speed = $4\sqrt{2}$ pixels / frame



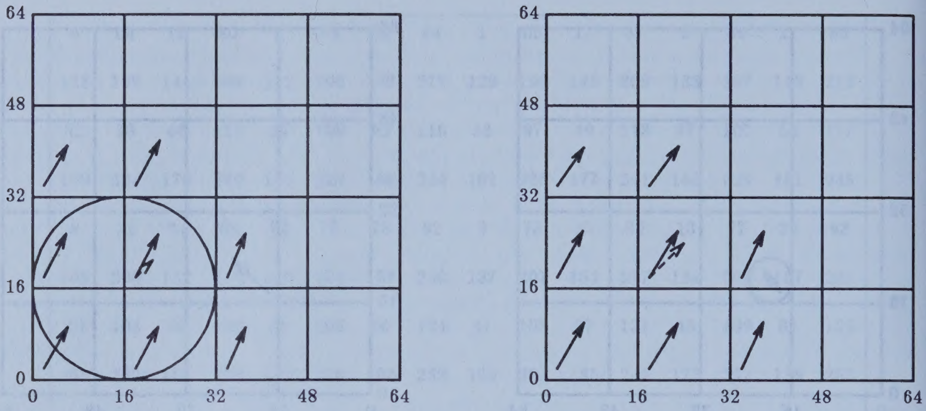
c

speed = $6\sqrt{2}$ pixels / frame

d

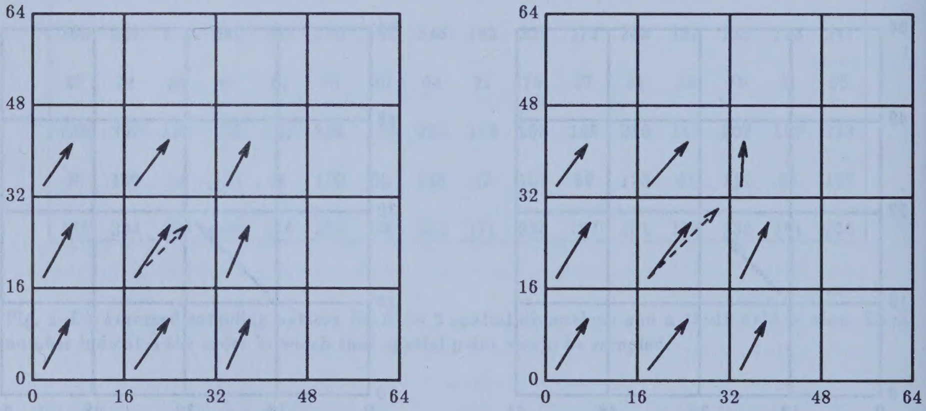
speed = $8\sqrt{2}$ pixels / frame

Fig. 2. Velocity field representations for a Gaussian object with radius $\rho=4$ undergoing constant speed translational motion at a 45 degree angle. The field of view is 64×64 and the blocks are 16×16 each. The dashed arrows represent the true velocity vectors. The circle represents the starting position of the moving object.



a
speed = $\sqrt{2}$ pixels / frame

b
speed = $2\sqrt{2}$ pixels / frame



c
speed = $3\sqrt{2}$ pixels / frame

d
speed = $4\sqrt{2}$ pixels / frame

Fig. 3. Velocity field representations for a Gaussian object with radius ρ=16 undergoing constant speed translationl motion at a 45 degree angle. The field of view is 64×64 and the bolcks are 16×16 each. The dashed arrows represent the true velocity vectors. The circle represents the starting position of the moving object.

FAST ITERATIVE IMAGE RESTORATION ALGORITHMS*

AGGELOS K. KATSAGGELOS and SERAFIM N. EFSTRATIADIS
Northwestern University
Department of Electrical Engineering and Computer Science
The Technological Institute, Evanston, Illinois 60208

ABSTRACT

In this paper a class of iterative image restoration algorithms is derived based on a representation theorem for the generalized inverse of a matrix. These algorithms exhibit a first or higher order of convergence and some of them consist of an "on-line" and an "off-line" computational part. The conditions for convergence and the rate of convergence of these algorithms are derived. A faster rate of convergence can be achieved by increasing the computational load. The algorithms can be applied to the restoration of signals of any dimensionality. Iterative restoration algorithms that have appeared in the literature represent special cases of the class of algorithms described here.

INTRODUCTION

The recovery or restoration of a signal that has been distorted is one of the most important problems in signal processing applications. For example, it is probably safe to say that there is no field where images are aquired that does not have active or potential work in image restoration.

More specifically the following degradation model is considered

$$y = Kx \tag{1}$$

where the vectors y, and x represent respectively the lexicographically ordered noisy blurred available image and the original image. The matrix K represents the space invariant or space varying deterministic distortion. The image restoration problem is then to invert Eq. (1) or to find an image as close as possible to the original one subject to a suitable optimality criterion, given y and K. The noise that is present in every image formation process is omitted from the model of Eq. (1) for the time being; it will be considered in a later part of the paper.

In general the matrix K in Eq. (1) is an $m \times n$ matrix, where $m \geq n$. That is, $K \epsilon L(R^n, R^m)$, $x \epsilon R^n$ and $y \epsilon R^m$, where $L(R^n, R^m)$ is the set of matrices that map R^n into R^m, the n-dimensional and m-dimensional Euclidean spaces, respectively. Let $R(K)$ and $N(K)$ denote respectively the range and the null space of K and $dim(S)$ the dimensionality of the subspace S [8]. If $dim(R(K)) = dim(R(K^T)) = r$, then $dim(N(K)) = n - r$ and $dim(N(K^T)) = m - r$, where ^{T} denoted the transpose of a matrix or a vector. Clearly Eq. (1) has at most one solution if and only if $r = n$, and we get no solution if $y \epsilon N(K^T)$. The degradation model of Eq. (1) can be modified so that K is a square matrix ($m = n$), by increasing the size of x by adding zeros or by reducing the size of y. Even in this case, however, for a large number of common distortions (motion blur, defocussing) the distortion matrix is singular, that is $r < n$. In both cases (square and rectangular K), it cannot be guaranteed that $y \epsilon R(K)$, therefore we want to be able to solve Eq. (1) for an "optimum" x.

The approach towards this direction is to solve for an x that minimizes

$$\| Kx - y \| \tag{2}$$

---

* This work was supported by the National Science Foundation under grant no. MIP-8614217.

where $||z||$ denotes the Euclidean norm of the vector z. Such an x is called a least squares solution (l.s.s) of Eq. (1). The l.s.s. satisfies the normal equations

$$K^T Kx = K^T y \tag{3}$$

The set of x's that satisfy Eq. (3) forms a closed convex set which contains a unique vector of minimal norm [1]. This vector is chosen to be the l.s.s. uniquely associated with y. Then the generalized inverse $K^+ \epsilon L(R^m, R^n)$ is defined by $K^+ y = x^+$, where x^+ is the least squares solution of minimal norm (l.s.s.m.n.) of Eq. (1).

Iterative algorithms are used in our work in solving the image restoration problem or in finding the l.s.s.m.s. of Eq. (1), due to their advantages over other existing restoration techniques [2,11]. The derivation of the iterative techniques will be based on a representation theorem for the generalized inverse K^+ of a matrix K to be presented in the following section. Based on this representation a number of iterative algorithms will be presented in obtaining K^+ as well as x^+, in section 3. The noise in the degradation model will be considered in section 4. Finally, experimental results will be presented.

REPRESENTATION OF K^+

A general theorem representing the generalized inverse of a matrix as the limit of a sequence of matrices presented in Groetch [1], is also presented here without proof due to its significance.

Theorem 1:
Suppose $K \epsilon L(R^n, R^m)$ and let $K^* = K^T K / R(K^T)$. If Ω is an open set with $\sigma(K^*) \subset \Omega \subset (0,\infty)$ and $\{S_\alpha(x)\}$ is a family of continuous real valued functions on Ω with $\lim_\alpha S_\alpha(x) = x^{-1}$ uniformly on $\sigma(K^*)$ then

$$K^+ = \lim_\alpha S_\alpha(K^*) K^T \tag{4}$$

where the convergence is in the uniform topology for $L(R^m, R^n)$. Furthermore,

$$||S_\alpha(K^*)K^T - K^+|| \leq sup \, |xS_\alpha(x) - 1| \quad ||K^+|| \tag{5}$$

where the supremum is taken over $x \, \epsilon \sigma(K^*)$.

Some of the notation used above is as follows: The spectral radius of a square matrix T and the restriction of T to a subspace S of R^n are respectively denoted by $\sigma(T)$ and T/S [8]. Clearly the matrix K^* is symmetric, positive definite, therefore its spectral radius is a subset of the set $(0,\infty)$ [7].

Theorem 1 is very powerful because it provides us with a general expression (Eq. (4)) for representing and iteratively computing the generalized inverse of a matrix. Furthermore it provides us with a measure of the rate of convergence (Eq. (5)). Therefore, it unifies (as will also become clear later) a large number of the iterative restoration techniques. Any family of functions $\{S_\alpha\}$ with the properties stated by Th. 1 can result to a new representation for K^+. Clearly some of these families of functions result in more attractive, from a computational point of view, representations. It is these functions $\{S_\alpha\}$ we are considering next. It is stated here that Th. 1 holds not only for matrices but for any linear operator with closed range [1].

ITERATIVE RESTORATION ALGORITHMS

Algorithm 1: Consider the sequence defined by

$$S_k(x) = \beta \sum_{i=0}^{k} (1-\beta x)^i \tag{6}$$

Clearly

$$\lim_{k \to \infty} S_k(x) = x^{-1} \tag{7}$$

uniformly on compact subsets of the set

$$\Omega_\beta = \{x : | 1-\beta x | < 1\} = \{x : 0 < x < \frac{2}{\beta}\} \tag{8}$$

It is easily shown [1,7] that $\sigma(K^*) \subset (0, \| K \|^2]$. Therefore, for

$$0 < \beta < 2 \| K \|^{-2} \tag{9}$$

$\sigma(K^*) \subset \Omega_\beta$. Application of Th. 1 results in

$$K^+ = \beta \sum_{i=0}^{\infty} [I - \beta K^T K]^i K^T \tag{10}$$

where I is the identity matrix. Now if we set

$$K_k = \beta \sum_{i=0}^{k} [I - \beta K^T K]^i K^T \tag{11}$$

then

$$K_0 = \beta K^T$$

$$K_{k+1} = (I - \beta K^T K) K_k + \beta K^T \tag{12}$$

The convergence of the sequence $\{K_k\}$ to K^+ is in the uniform topology for $L(R^m, R^n)$. Uniform convergence guarantees strong convergence [10], therefore by multiplying both sides of iteration (12) by y and by setting $x_k = K_k y$ we get

$$x_0 = \beta K^T y$$

$$x_{k+1} = (I - \beta K^T K) x_k + \beta K^T y$$

$$= x_k + \beta K^T (y - K x_k) \tag{13}$$

Iteration (13) also results from a successive approximations approach to the solution of the normal equations (Eq. (3)). Iteration (13) has been studied and used extensively for signal restoration [2,3,11]. When the distortion is space invariant then one- or two-dimensional convolutions can replace the matrix vector multiplications.

Error estimate: With the use of Eq. (5) it is shown that [1,7]

$$||K_k - K^+|| \leq ||K^+|| \, c^{k+1} \tag{14a}$$

or

$$\frac{||x_k - x^+||}{||x^+||} \leq c^{k+1} \tag{14b}$$

where

$$c = \max\{ |1-\beta||K||^2|, |1-\beta||K^+||^{-2}| \} \tag{15}$$

and x^+ denoted the minimum norm least squares solution of Eq. (1). According to Eq. (14b) the error converges linearly to zero with a convergence factor c.

Algorithm 2: x^{-1} can be regarded as the root of the function $S(y) = y^{-1}-x$. If the Newton-Raphson method is used in approximating this root, then the sequence y_k is generated according to

$$y_{k+1} = y_k \, (2-xy_k) \tag{16}$$

for suitable y_0. Suppose that for $\beta>0$ a sequence of functions $\{S_k(x)\}$ is defined by

$$S_0(x) = \beta$$

$$S_{k+1}(x) = S_k(x)(2-xS_k(x)) \tag{17}$$

The sequence defined by (17) is shown [1,7] to converge to x^{-1} uniformly on compact subsets of Ω_β defined by Eq. (8). Therefore, according to Th. 1 the sequence of operators $\{S_k(K^*)\} \subset L(R(K^T), R(K^T))$ defined by

$$S_0(K^*) = \beta I$$

$$S_{k+1}(K^*) = S_k(K^*)(2I - K^T K S_k(K^*)) \tag{18}$$

has the property that $\lim_k S_k(K^*)K^T = K^+$, uniformly in $L(R^m, R^n)$, for β satisfying (9). By setting $K_k = S_k(K^*)K^T$, we get

$$K_0 = \beta K^T$$

$$K_{k+1} = K_k(2I - KK_k) \tag{19}$$

Again since strong convergence is guaranteed by setting $x_k = K_k \, y$ we get the vectorwise version of iteration (19), that is

$$x_0 = \beta K^T y$$

$$x_{k+1} = (2I - K_k K)x_k \tag{20}$$

Iteration (20) can be decoupled by setting $K_k^\# = K_k K$. Then from Eqs. (19) and (20) we get

$$x_0 = \beta K^T y \tag{21a}$$

$$x_{k+1} = (2I - K_k^{\#}) x_k \tag{21b}$$

$$K_{k+1}^{\#} = (2I - K_k^{\#}) K_k^{\#} \tag{21c}$$

The advantage of the form (21a,b,c) is that $K_k^{\#}$ can be computed in advance or off-line. Then Eq. (21b) is the on-line part. As observed from Eq. (21c) the limit of $K_k^{\#}$ is the projection onto the row space of K. When K has an inverse then the limit of $K_k^{\#}$ is the identity. According to Eq. (21c) the distortion operator is also updated; since the signal x_k is restored at each iteration, the distortion operator that maps the original signal into x_k is approaching the identity operator (if the inverse exists) as the iteration number increases. Iteration (21) was derived by Singh et al. [4] only for the case that K is invertible. Clearly, our derivation generalizes the previously reported results.

Error estimate: It has been derived [1,7] that for algorithm 2 the rate of convergence is quadratic, that is

$$||K_k - K^+|| \leq ||K^+|| c^{2^k} \tag{22a}$$

or

$$\frac{||x_k - x^+||}{||x^+||} \leq c^{2^k} \tag{22b}$$

where the convergence factor c is given by Eq. (15).

Algorithm 3: The following methods are called hyperpower methods. They are extrapolating on the desirable quadratic convergence property of the previous iteration and they are displaying p-th order of convergence, where $p \geq 2$ is an integer.

Given an integer $p \geq 2$, the functions $\{S_k^p(x)\}$ are defined by

$$S_0^p = \beta > 0$$

$$S_{k+1}^p(x) = S_k^p(x) \sum_{i=0}^{p-1} (1 - x S_k^p(x))^i \tag{23}$$

Note that for $p = 2$ Eq. (23) agrees with Eq. (17). It is shown [1,7] that the sequence defined by Eq. (23) converges uniformly to x^{-1} on compact subsets of Ω_β of Eq. (8). Therefore, according to Th. 1 the sequence of operators $\{S_k^p(K^*)\} \subset L(R(K^T), R(K^T))$ defined by

$$S_0^p = \beta I$$

$$S_{k+1}^p(K^*) = S_k^p(K^*) \sum_{i=0}^{p-1} (I - K^T K S_k^p(K^*))^i \tag{24}$$

has the property that $\lim_k S_k^p(K^*) K^T = K^+$, uniformly in $L(R^m, R^n)$ for β satisfying Eq. (9). Since [7]

$$S^p_{k+1}(K^*)K^T = S^p_k(K^*)K^T \sum_{i=0}^{p-1} (I-KS^p_k(K^*)K^T)^i$$

by setting $K_k = S_k(K^*)K^T$, we get

$$K^p_0 = \beta K^T$$

$$K^p_{k+1} = K^p_k \sum_{i=0}^{p-1} (I-KK^p_k)^i \tag{25}$$

The vectorwise version of Eq. (25) is derived to be

$$x^p_0 = \beta K^T y$$

$$x^p_{k+1} = \sum_{i=0}^{p-1} (I-K^p_k K)^i x^p_k \tag{26}$$

Again Eq. (25) may comprise the off-line while Eq. (26) the on-line computational part.

Error estimate: In a similar way to algorithm 2, it is shown [1,7] that algorithm 3 has p-th order of convergence, that is

$$||K^p_k - K^+|| \le ||K^+|| c^{p^k} \tag{27a}$$

or

$$\frac{||x_k - x^+||}{||x^+||} \le c^{p^k} \tag{27b}$$

where the convergence factor is given by Eq. (15).

Algorithm 4: The previous iterations were based on approximating the function x^{-1} and then representing these approximations via the spectral theorem. Similarly the generalized inverse can be approximated by interpolating the function x^{-1} at certain integral values and using Th. 1. Then it is shown [1,7] that

$$K_0 = K^T$$

$$K_{k+1} = K_k + \frac{1}{k+2}[K^T - K^T KK_k] \tag{28}$$

converges to K^+ in the uniform topology for $L(R^m, R^n)$. The vectorwise version of (28) then becomes [7]

$$x_0 = K^T y$$

$$x_{k+1} = x_k + \frac{1}{k+2}K^T[y - Kx_k] \tag{29}$$

Iteration (29) relates the the method of stochastic approximations [9]. The error estimate is somewhat more complicated for algorithm (28)-(29); the convergence factor is now a function of the iteration index k [1,7].

REGULARIZED ITERATIVE RESTORATION ALGORITHMS

So far the additive noise in the degradation model (Eq. (1)) has been ignored. A more realistic degradation model is

$$y = Kx + n \tag{30}$$

where n represents the white Gaussian noise. The restoration problem then is to find an image as close as possible to the original image x subject to a suitable optimality criterion. The restoration problem is an ill-posed inverse problem, which means that the noise in the data is amplified in the solution. Regularization techniques are used in controlling the trade-off between noise amplification and sharpness in the restored image. We used [3,11] one such regularization technique due to Miller. The restored image is now the solution of

$$(K^T K + \alpha C^T C)x = K^T y \tag{31a}$$

or

$$Ax = b \tag{31b}$$

where the constraint C incorporates knowledge about the original signal into the iteration and the regularization parameter α depends on the signal to noise ratio and needs to be adjusted [3,11]. According to Eq. (31b) when noise is present in the data the restoration problem is again the solution of a linear set of equations for the minimum norm least squares solution or for the generalized inverse of A. This means that all the algorithms presented in section 3 can be used by replacing K with A from Eq. (31b). However, since A is a square positive semidefinite matrix, we want in the case that Eq. (31a) has a solution, to find the minimum norm solution, without considering the least squares approach. Towards this end we have shown [11] that the iteration

$$x_{k+1} = x_k + \beta(b - Ax_k) \tag{32}$$

converges to the minimum norm solution of (31b) plus the projection of the initial condition onto the null space of A, for $0<\beta<||A||^{-1}$, if and only if $b \in R(A)$. Clearly iteration (32) can be used in solving Eq. (1) if K is positive semidefinite and the rest of the conditions stated above apply.

If we think of iterations (13) and (32) as forming a corresponding pair since they both have linear rate of convergence, we now propose the corresponding algorithms to algorithms 2 and 3. These algorithms have p order of convergence, where $p \geq 2$, and solve for the minimum norm solution of Eq. (31b) when it exists, without following a least squares approach. Towards this end we have proved [7] the following theorem.

Theorem 2: Let $A : R^n \to R^n$ be a positive semidefinite matrix. For $A_0 = \beta A$ and $x_0 = \beta b$ consider the iterative process

$$x_{k+1} = \sum_{i=0}^{p-1} (I - A_k)^i x_k \tag{33a}$$

$$A_{k+1} = \sum_{i=0}^{p-1} (I - A_k)^i A_k \tag{33b}$$

where $0<\beta<||A||^{-1}$. Then the sequence $\{x_k\}$ converges to the minimum norm solution of Eq. (31b), if and only if $b \epsilon R(A)$. Iteration (33) has p-th order of convergence, as discussed in algorithm 3. Algorithm (33) has been also discussed by Morris et. al. [5,6] for the case of linear shift invariant distortion by using a frequency domain representation. In their work, convergence was shown only for the case that A is invertible. Theorem 2 extents their results.

EXPERIMENTAL RESULTS

Algorithms 1, 2 and 3 with $p=3,5$ are used in removing a linear shift-invariant distortion from the original signal $x(n)=\delta(n-30)+\delta(n-35)$. The impulse response of the distorting system is equal to

$$k(n)=\begin{cases}\dfrac{1}{2N+1}, & n=-N,..,0,..,N\\ 0, & otherwise\end{cases} \tag{34}$$

with N=5. In this case the matrix K is circulant with eigenvalues the Discrete Fourier Transform (DFT) values of $k(n)$. Since certain DFT values are zero, the matrix K is singular. The original signal $x(n)$ and the distorted signal $y(n)=x(n)*k(n)$, where * denotes convolution, are shown in Fig. 1. The parameter β was set to be equal to 1.9 in all the experiments. Algorithms 1, 2 and 3 with $p=3$ and $p=5$ are used in restoring $y(n)$. The restored signal after 9 iterations is shown respectively in Figs. 2a,b,c,d. Clearly, convergence has been reached for algorithm 3 with $p=5$, as is shown in Fig. 2d. The norm of the difference between successive iterations normalized by the total number of samples in the sequence, that is, $||x_{k+1}-x_k||/M$, is shown in Fig. 3a, as a function of the number of iterations. It is of interest to observe that this norm is not a monotonically decreasing function. Finally, the normalized residual error $||x_k-x_{or}||/||x_{or}||$, is shown in Fig. 3b as a function of the number of iterations. The number of iterations required by each algorithm to reach convergence is apparent from this plot. These numbers agree with the values obtained by considering the error estimates described by Eqs. (14b), (22b) and (27b). That is, the 9 iterations with p=5, correspond to 10 iterations with p=4, 13 iterations with p=3, 20 iterations with p=2 and about 2 million iterations with algorithm 1.

DISCUSSION AND CONCLUSIONS

Based on a representation theorem for the generalized inverse of a matrix a number of iterative signal restoration algorithms have been derived. Some of these algorithms extent the results previously derived. In this paper the words signal and image have been used interchangeably. In fact the algorithms presented can be used for the restoration of a signal of any dimensionality as well as for the solution of any type of inverse problem which accepts the formulation of Eq. (1) or Eq. (30). When dealing with images spatially adaptive restoration algorithms have been derived [11] which incorporate into the restoration properties of the human visual system. The algorithms presented can be also used in this case.

In all the vectorwise iterative algorithms presented hard constraint [2,11] can be used, which incorporate into the algorithm a priori knowledge about the original signal. Certain of these hard constraints can be implemented as projections onto convex sets. Clearly the convergence conditions, the limiting solution and the rate of convergence may be different in this case. Our experimental results suggest that the use of hard constraints in algorithm 3 may result in undesirable solutions. This is due to the independent calculation of K_k and x_k.

An issue to be further investigated is the comparison of the algorithms presented. This comparison can be based on the trade-off between computational load and convergence rate and on the sensitivity of each algorithm to noise (use of Eq. (30) with no regularization). The algorithms with the off line computational part may be eventually used for real time restoration. Finally, another interesting issue to be investigated is the use of other families of functions $\{S_\alpha(x)\}$ which satisfy Th. 1, that may lead to useful iterative restoration algorithms.

REFERENCES

1. C. W. Groetch, *Generalized Inverses of Linear Operators,* Marcel Dekker, Inc., 1970.

2. R. W. Schafer, R. M. Mersereau and M. A. Richards, "Constrained Iterative Restoration Algorithms," *Proc. IEEE,* vol. 60, pp. 432-450, April 1981.

3. A. K. Katsaggelos, J. Biemond, R. M. Mersereau and R. W. Schafer, "A General Formulation of Constrained Iterative Restoration Algorithms," *Proc. 1985 Int. Conf. Acoust., Speech, Signal Processing,* pp. 700-703, Tampa FL, March 1985.

4. S. Singh, S. N. Tandon and H. M. Gupta, "An Iterative Restoration Technique," *Signal Processing 11,* pp. 1-11, 1986.

5. C. E. Morris, M. A. Richards and M. H. Hayes, "Iterative Deconvolution Algorithm with Quadratic Convergence," *J. Opt. Soc. Am. A.,* vol. 4, no. 1, pp. 200-207, Jan. 1987.

6. C. E. Morris, M. A. Richards and M. H. Hayes, "A Generalized Fast Iterative Deconvolution Algorithm," *Proc. 1987 Int. Conf. Acoust., Speech, Signal Processing,* pp. 36.1.1-36.1.4, Dallas, TX, April 1987.

7. A. K. Katsaggelos, *A Unified Approach to Iterative Signal Restoration,* Technical Report IPL-87-2, Northwestern University, Department of Electrical Engineering and Computer Science, June 1987.

8. G. Strang, *Linear Algebra and its Applications,* 2nd edition, Academic Press, New York, 1980.

9. H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems,* Applied Math. Sciences, vol. 26, Springer-Verlag, 1978.

10. A. W. Naylor and G. R. Sell, *Linear Operator Theory in Enginnering and Science,* Springer-Verlag, 2nd edition, 1982.

11. A. K. Katsaggelos, J. Biemond, R. W. Schafer and R.M. Mersereau, "Constrained Iterative Image Restoration Algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* submitted for publication, Dec. 1986.

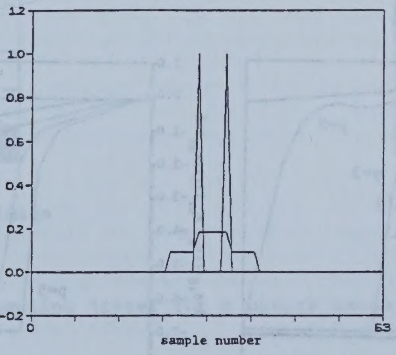


Figure 1: Original and distorted signal.

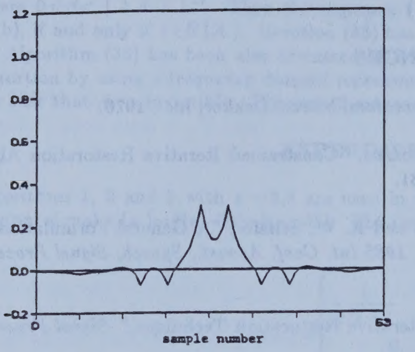


Figure 2a: Restored signal by algorithm 1 after 9 iterations.

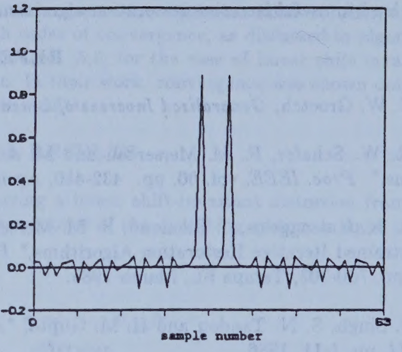


Figure 2b: Restored signal by algorithm 2 after 9 iterations.

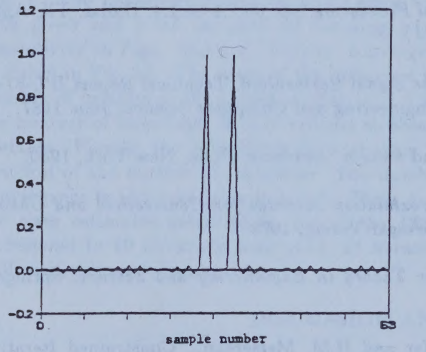


Figure 2c: Restored signal by algorithm 3; p=3, 9 iterations.

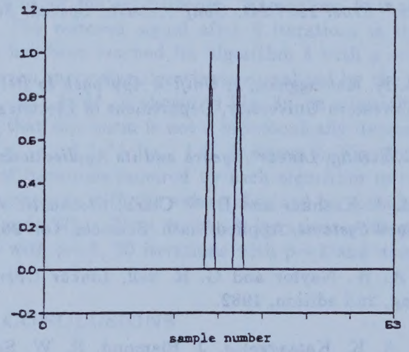


Figure 2d: Restored signal by algorithm 3; p=5, 9 iterations.

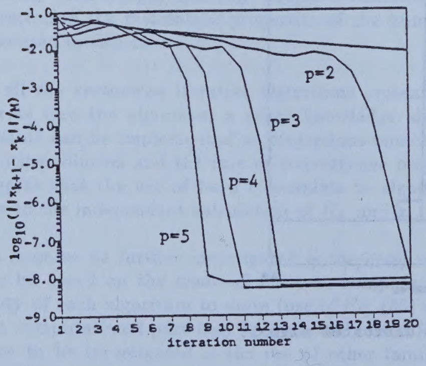


Figure 3a: Error between successive iteration steps.

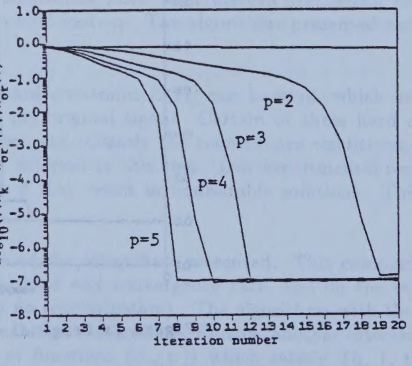


Figure 3b: Normalized residual error.

RESTORATION OF IMAGES DISTORTED BY SAMPLING JITTER

PINGNAN SHI & RABAB KREIDIEH WARD
Department of Electrical Engineering
University of British Columbia
Vanconver B.C. Canada V6T 1W5

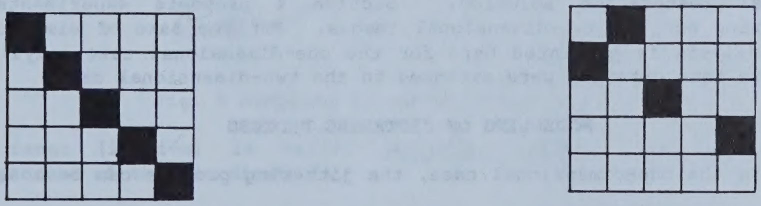
ABSTRACT

Images sampled with mechanical scanners suffer from sampling or position jitter noise. For sampling jitter with a random probability characteristics, a restoration technique is presented. The restoration technique minimizes the expected value of the error squared. The computations are done in the space domain via a sliding window and are thus easy to implement.

INTRODUCTION

Since computers can process only digital images, a neccessary precusor to digital image processing is the conversion of physical images into digital forms. To accomplish this, an image digitizer is needed to spatially sample points of a physical image. Two relevent components of a digitizer are: sampling aperture and scanning mechanism. A sampling aperture allows the digitizer to access picture elements (pixels) individually while ignoring the remainder of the image. A scanning mechanism moves the sampling aperture over the image in a predetermined pattern. Scanning allows the sampling aperture to address pixels in order, one at a time.

During sampling, ideally the physical image is linearly sampled. However, imperfection in the scanning mechanism might lead the sampling aperture to the wrong point. For instance, a physical image shown in Fig. 1 could be digitized into such a digital image as shown also in Fig. 1 due to the imperfection in the scanning mechanism.



(a) Original image (b) Jittered image

Fig. 1 Sampling jitter for a binary image

In Fig. 1 each square represents a pixel. Note that the dark spots in the physical image are jittered around a line in the digital image. This kind of distortion is called sampling jitter, which causes salt-and-pepper like noise and random geometrical distortion in the output image.

Similar processes with similar noisy effects occur in digital image display [1]. This is refereed to in the literature as spot position noise and is caused by noise in the display spot deflection circuit. Display position noise causes non-uniformity in display spot spacing. Another example of recent concern [2] is the digitization of astronomical plates. Here although extremely accurate equipments are used, irregularities in the sampling grid due to the mechanical scanning system are unavoidable. The effects of some wobbling and to-and-fro movement of the carriage (backlash) can be easily seen in the digitized image.

The objective of restoration of images distorted by sampling jitter is to generate an image as close as possible to the original one based on 1) the knowledge of the jittered image, 2) probability chraracteristics of the jittering process, and 3) information about the original image if available.

Sampling jitter is rarely mentioned in tutorial materials in the area of image restoration. This unpopularity probably results from the experience that in practice, sampling jitter does not cause as severe distortions as those caused by other sources of distortions such as additive noise or motion blur. However, restoration of jittered images provides a challenge of academic interests since it is essentially different from conventional distortions and new analytical tools might be required or developed in the study of this subject.

This paper analyzes the jittering process and derives an optimal solution for restoration of jittered images. Experiments on restoring one- and two-dimensional images are also presented in this paper. Section 2 discusses the jittering process. Section 3 derives the optimal restoration solution. Section 4 presents experiments of restoring one and two-dimensional images. For the sake of simplicity, the analysis is presented here for the one-dimensional case only. The results thus obtained were extended to the two-dimensional case.

MODELLING OF JITTERING PROCESS

In the one-dimensional case, the jittering process can be modelled as,

$$g_i = f_{(i-n(i))} \tag{1}$$

where

f_i represents the grey level of pixel at position i of the original image;

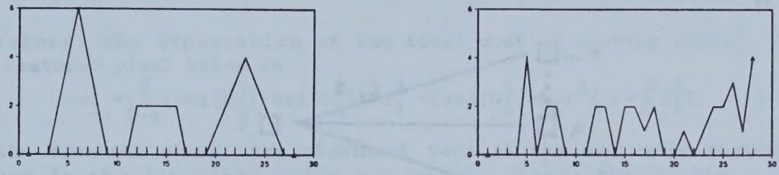
g_i represents the grey level of pixel at position i of the jittered image;

n(.) is a random function representing the sampling jitter;
i is the index (integer).

n(.) is assumed here as a stationary random function with zero mean

$$E\{n(.)\}=0 \tag{2}$$

An example of a one-dimensional image degraded by sampling jitter is shown in Fig. 2.



(a) Original image (b) Jittered image

Fig. 2 An example of one-dimensional jittering

Note that in equation (1), the noise is in the coordinate i. Assume that the width of the noise n(i) extends at position i over a range, i.e.

$$-m<n(i)<m \tag{3}$$

then the distorted pixel g_i can take value from $f_{(i-m)}$ to $f_{(i+m)}$ (see Fig. 3).

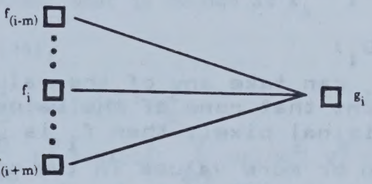


Fig. 3 Sampling jitter at position i.

The range [i-m,i+m] is called jittering range. Its length is independent of the coordinate i.

Since the jittered image has the same size as the original one, any redundancy of values from the original image to the jittered one causes at the same time loss of some other values, i.e., some values of the original image may never appear in the jittered image. Therefore, some information of the original image is lost during the jittering process. This implies that we can not obtain a perfectly restored image (exactly the same as the original image).

RESTORATION OF JITTERED IMAGE

Suppose we are restoring the pixel at position i. Then, what we want to know is the value of the original pixel at position i. From equation (3) we know that only pixels of $g_{(i-m)}$ to $g_{(i+m)}$ can take the value of the original pixel f_i. Therefore, a relationship between the restored pixel and the pixels within the jittering range has to be established in order to find the restored pixel value (grey level) (See Fig. 4).

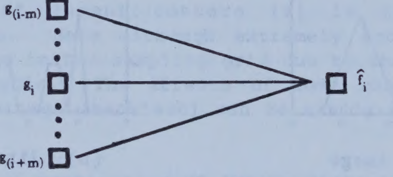


Fig. 4 Restoration of the ith pixel

where $\hat{f}_i$ denotes the restored grey level of the ith pixel. The pixels of $g_{(i-m)}$ to $g_{(i+m)}$ forms a set and this set is denoted as G_i.

$$G_i = \{g_{(i-m)}, \ldots, g_i, \ldots, g_{(i+m)}\} \tag{4}$$

G_i is called the ith pixel set.

The estimate $\hat{f}_i$ in the least mean square sense is the answer to

$$\underset{\hat{f}_i}{\text{Minimize}} \quad E\{(\hat{f}_i - f_i)\} \tag{5}$$

$$\text{such that} \quad \hat{f}_i \in \{G_i\} \tag{6}$$

The unknown value f_i can take any of the values g_i in the set $\{G_i\}$. In the event that none of the values in $\{G_i\}$ took the value of the original pixel, then f_i is unrecoverable.

In case any two or more values in the set $\{G_i\}$ are equal, let us define the set $\{\text{val}[G_i]\}$ to be the set which contains the unrepeated values in $\{G_i\}$ only, i.e.,

$$\text{val}[G_i] = \{\text{val}[G_i^0], \text{val}[G_i^1], \ldots, \text{val}[G_i^n]\} \tag{7}$$

where $n+1 \leq 2m+1$.

Let P_i^j the posterior probability of f_i, be

$$P_i^j = \Pr\{f_i = \text{val}[G_i^j] \mid \text{observation } \{G_i\}\}.$$

In the appendix we obtain P_i^j in terms of the known probabilities $P_{(k-i)}$, where $P_{(k-i)} = Pr(g_k = f_i)$. Since the random function n(.) is assumed stationary, the probability of $g_k = f_i$ is a function of the relative distance between g_k and f_i in position, i.e., $Pr(g_k=f_i) = Pr(g_{(k+m)}=f_{(i+m)})$. Chosing any value in the set $val[G_i]$ to be the restored pixel value has to pay a price when the event of it being the original pixel value is not true. The cost of chosing $val[G_i^j]$ as the restored pixel value when $val[G_i^l]$ is the original pixel value is

$$\varepsilon_i^{j,\ell} = (val[G_i^j]-val[G_i^\ell])^2 \qquad (8)$$

Therefore, the expectation of the total cost of chosing $val[G_i^j]$ to be the restored pixel value is

$$\varepsilon_i^j = \sum_{\ell=0}^{n} (val[G_i^j]-val[G_i^\ell])^2 P_i^\ell + (val[G_i^j]-f_i)^2 (1-\sum_{\ell=0}^{n} P_i^\ell) \qquad (9)$$

In the equation above, the rightmost term is the cost when neither of values in the ith pixel value set $val[G_i]$ takes the original pixel value.

Since f_i is not known, the rightmost term in equation (9) has to be neglected from the total cost in order to get a practical answer.
As long as f_i is not too far away from values in the set $val[G_i]$, because $(1-\sum_{\ell=0}^{n} P_i)$ is very small, this neglect is justifiable.
Therefore, the alternative total cost of chosing $val[G_i]$ to be the restored pixel value is

$$\varepsilon_i^j = \sum_{\ell=0}^{n} (val[G_i^j]-val[G_i^\ell])^2 P_i^\ell \qquad (10)$$

Our objective is to find a pixel value which has the minimal cost in equation (10). Try every pixel value in the value set $val[G_i]$, the one which yields the minimal cost is chosen as $\hat{f}_i$

From equation (10),

$$\begin{aligned} \varepsilon_i^j &= \sum_{\ell=0}^{n} (val(G_i^j]-val[G_i^l])^2 P_i^\ell \\ &= val^2[G_i^j] \sum_{\ell=0}^{n} P_i^\ell - 2val[G_i^j] \sum_{\ell=0}^{n} val[G_i^\ell] P_i^\ell + \sum_{\ell=0}^{n} val^2[G_i^\ell] P_i^\ell \qquad (11) \end{aligned}$$

Since the third term is constant for all $j \in \{0,1,...,n\}$, the equivalent cost function for the optimal solution search is,

$$\begin{aligned}\varepsilon_i^j &= \mathrm{val}^2[G_i^j] \sum_{\ell=0}^{n} P_i^\ell - 2\mathrm{val}[G_i^j] \sum_{\ell=0}^{n} \mathrm{val}[G_i^\ell]P_i^\ell \\ &= \mathrm{val}[G_i^j](\mathrm{val}[G_i^j] \sum_{\ell=0}^{n} P_i^\ell - 2 \sum_{\ell=0}^{n} \mathrm{val}[G_i^\ell]P_i^\ell) \qquad (12)\end{aligned}$$

The optimal solution for equation (12) may not be unique. In the set $\mathrm{val}[G_i]$, there may be another pixel value which can be the optimal solution. Based on the assumption that an image is continuous within a small neighborhood, the value which is closer to the value of the previously restored pixel should be chosen as the restored pixel value.

The jittering range may not be finite as in the case when n(.) is Gaussian. Even if the jittering range is finite, if it is very large, heavy computation is required. To save computing time, the optimal solution search should be limited within a smaller range. Thus, the concept of a processing window is introduced. A processing window has the position of the processed pixel as its center, and its size is chosen such that the probability of the original pixel value being jittered outside the window is small enough to neglect.

TESTS

In this section, we present two experiments, the first one is a one-dimensional image and the second one is a two-dimensional image. The one-dimensional image is shown in Fig. 2. The two-dimensional image is the face of the 'image girl', and is shown in Fig. 6.

The one-dimensional case

The original image is shown in Fig. 2. It is jittered by Gaussian noise with standard deviation of 2.0. The jittered image is also shown in Fig. 2. Two processing windows with different sizes of 3 and 5 are used in the restoration. The restored images are shown in Fig. 5. Note that in the jittered image there are eight peaks while in the original one there are only three. In the restored image, there are three peaks. Therefore, the essential information contained in the original image is recovered.

By the RMS (root mean square error) measurement, the RMS for the jittered image is 3.48%, the RMS for the restored image with window of size 3 is 3.39%, and the RMS for the restored image with window of size 5 is 3.05%. Eventhough, the reduction of RMS between the jittered image (Fig. 2(b)) and the restored image (Fig. 5(c)) is not significantly large, the visual effects of these two images are dramatically different.

The two-dimensional case

The original image is the face of the 'image girl'. It is jittered by Gaussian noise with standard deviation of 1.0. The white dots along the picture borders are the results when the sampler is jittered outside the image range. Two processing windows of size 3x3 and 5x5 are chosen for the restoration. The original, jittered and restored images are all shown in Fig. 6.

The RMS for jittered image is 8.74%, the RMS for the restored image with window of size 3x3 is 5.40%, and the RMS for the restored image with window of size 5x5 is 4.82%.

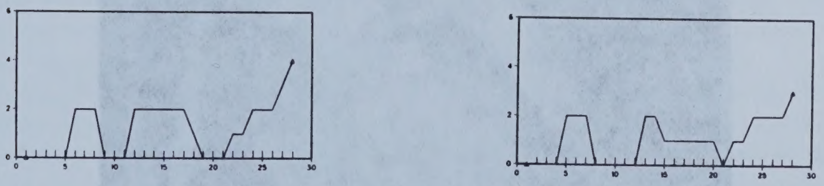
CONCLUSION

The results of the tests above show that our method works satisfactorily. The advantages of our method are: 1) easy to implement; and 2) no requirement on the size of the image to be processed. The computation required by our method has a maximum number of operations of $0[MxNxL^4]$, where M and N denote the length and the width respectively of the image to be processed, and L denotes the width of the processing window. The computation is carried out in spatial domain, thus the length and the width of the image is not restrained to be 2^k. Further improvement could be achieved by exploiting the non-stationary nature of images and by more accurate modelling of the jitteirng process.

REFERENCE

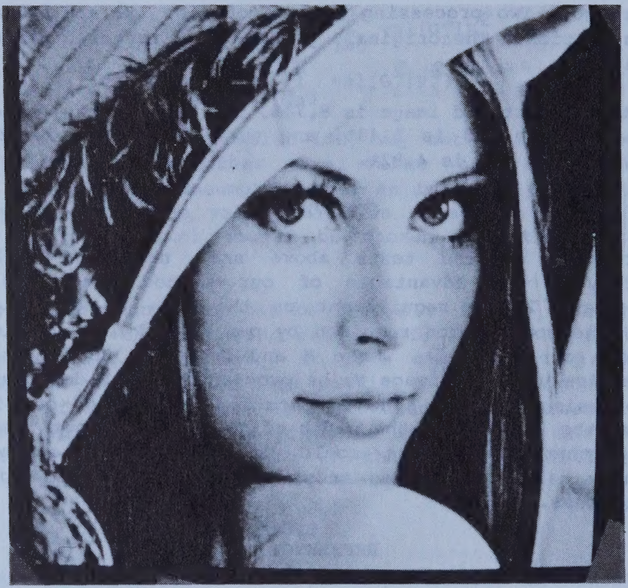
[1] K.R. Castleman, "Digital Image Processing," Prentice Hall, New Jersey, 1979, pp. 48-49.

[2] M. Piacentini, C. Cafforio, and F. Rocca, "Algorithms for image reconstruciton after non-uniform sampling," IEEE Trans on A.S.S.P., vol. ASSP-35, No. 8, August 1987, pp. 1185-1189.

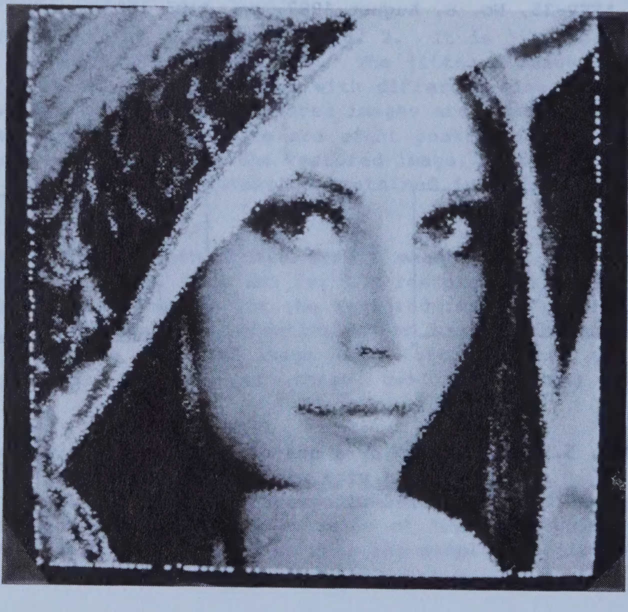


(a) (b)

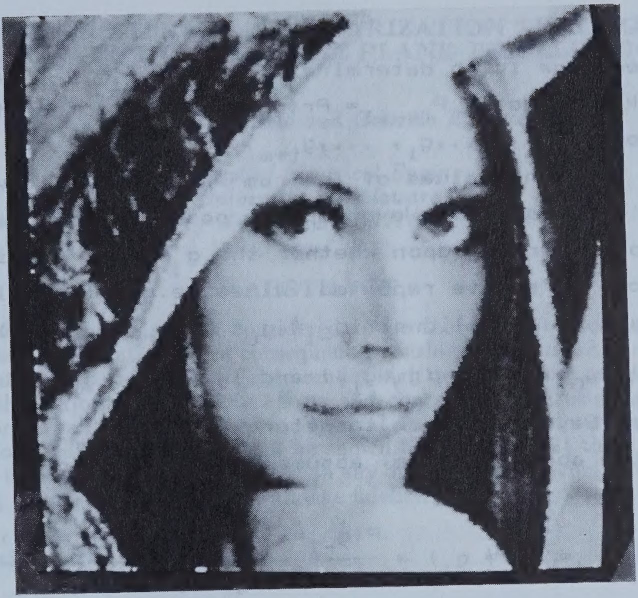
Fig. 5. Restoration of a one-dimensional image.
(a) restored with window of size 3;
(b) restored with window of size 5.



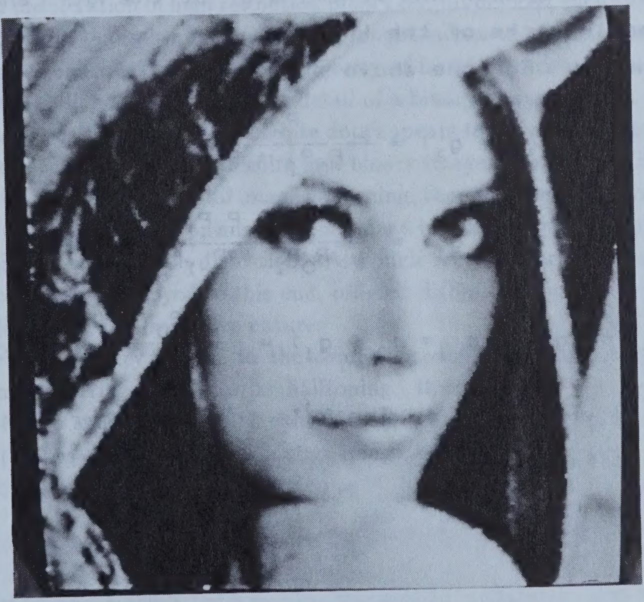
(a)



(b)



(c)



(d)

Fig. 6. Restoration of a two-dimensional image.
(a) Original;
(b) jittered;
(c) restored with window of size 3x3;
(d) restored with window of size 5x5

APPENDIX

The problem here is to determine P_i^j the $\Pr(f_i = g_j \mid \{G\})$ from the knowledge of $P_{(k-i)} = \Pr(g_k = f_i)$ and from the observation $(g_{i-m}, \ldots, g_i, \ldots, g_{i+m})$.

We assume that the values of f_i from which the g_j's were generated are distinct. We group the possible observations into groups depending upon whether the g_j's are distinct or whether some g_j's have repeated values, e.g., for m=1 the groups would be as follows: $(g_{-1} \neq g_0 \neq g_1)$, $((g_{-1} = g_0) \neq g_1)$, $((g_{-1} = g_1) \neq g_0)$, $(g_{-1} \neq (g_0 = g_1))$ and $(g_{-1} = g_0 = g_1)$.

Then we use Bayes' theorem to determine P_i^j.

Thus for the above example, assume that we find $g_{-1} = g_0 (\neq g_1)$, then

$$p(f_0 = g_j \mid g_{-1} = g_0 \neq g_1) = \frac{P(g_{-1} = g_0 \neq g_1 \text{ and } f_0 = g_j)}{P(g_{-1} = g_0 \neq g_1)}$$

Each of the numerator and denominator of the RHS can be determined in terms of the known $P_{(k-i)}$'s and thus P_i^j can be determined. For the above example we find that

$$p(f_0 = g_{-1} \mid g_{-1} = g_0 \neq g_1) = \frac{P_{-1} P_0 (1 - P_1)}{P_0 P_1 + P_{-1} P_0 (1 - P_1)}$$

$$p(f_0 = g_1 \mid g_{-1} = g_0 \neq g_1) = \frac{P_0 \; P_1 P_1}{P_0 P_1 + P_{-1} P_0 (1 - P_1)}$$

$$p(f_0 \neq (g_{-1} \text{ or } g_1) \mid g_{-1} = g_0 \neq g_1) = \frac{P_0 \; P_1 (1 - P_1)}{P_0 P_1 + P_{-1} P_0 (1 - P_1)}.$$

Note that $P_{-1} + P_0 + P_1 = 1$. The P_i^j's for the other groups are similarly determined.

VECTOR DITHERING: BINARIZATION THROUGH UNCONVENTIONAL IMAGE PLANE TESSELLATION

Tandhoni S. Rao and Gonzalo R. Arce

Department of Electrical Engineering
University of Delaware
Newark, DE 19716

ABSTRACT

Ordered dither has long been considered to be the simplest and most convenient method of image binarization. As a consequence, considerable research has been done to optimize the dithering process in order to render high quality binary images. This paper differs from the majority of previous efforts and concentrates on using unconventional tilings to improve halftone image quality. Optimal dither patterns for several representative tessellations will be derived. It will be shown by example that these patterns have superior gray scale rendition capabilities than any other previously considered dither pattern resulting in substantially improved image fidelity.

I. Introduction

The recent proliferation of high resolution binary output devices, such as laser printers, scientific workstations and plasma panels, has brought increased attention to the problem of optimal image binarization. The vast majority of image binarization techniques rely on the viewers inability to resolve the fine detail of a binary image; hence a well designed pattern of closely positioned black and white dots appears to be a uniform gray. A poorly designed pattern scheme, however, results in a binary image with clearly visible textures and patterns. The problem of digital image halftoning, therefore, becomes one of designing an algorithm which resolves a continuous tone image into a binary one while minimizing the introduction of perceivable distortion. Although recursive error diffusion methods have been successfully employed to this end, ordered dither techniques are preferred due to their simplicity and non-recursive nature.

Ordered dither is very similar to the contact screening methods long used in the graphic arts and printing industries for halftoning. It consists of thresholding the continuous tone image against a periodic spatial screen, known as the halftone cell. More formally, if $f(x,y)$ is the original continuous image and $f_{ij} = f(iR, jR)$ its samples, the discrete halftone image $h(x,y)$ can be written as

$$h(x,y) = \sum_{m} \sum_{n} p[m,n;f_{mn}] rect[(x-mR)/R] rect[(y-nR)/R], \tag{1}$$

where $rect(x) = 1, \ \ |x| < \frac{1}{2}$ and $0, \ \ |x| > \frac{1}{2}$. The function $p[m,n;f_{mn}]$ is the discrete dot profile which satisfies the bi-level absorptance constraint ie. $p[m,n;f_{mn}] = 0 \ or \ 1$. The binary value of the discrete dot profile is determined by

$$p[m,n;f_{mn}] = \begin{cases} 1, & \text{if } f_{mn} \geq \sigma_{mn}; \\ 0, & \text{else.} \end{cases} \tag{2}$$

The screen function σ_{mn} is two dimensionally periodic in a fashion explicitly dependent on the type of image plane tessellation employed.

The distribution of thresholds within a halftone cell plays a crucial role in the performance of ordered dither halftoning which can not be overemphasized. Different threshold distributions result in different dither patterns which in turn yield binary images of varying quality. With this in mind, several authors have considered the problem of choosing a dither pattern within a halftone cell which minimizes pattern visibility while retaining image detail [1]-[4].

This paper differs from previous efforts and concentrates on using unconventional tilings to improve halftone image quality. Optimal dither patterns for several representative tessellations will be derived. It will be shown by example that these patterns have superior gray scale rendition capabilities than any other previously considered dither pattern resulting in substantially improved image fidelity.

Before we proceed, however, a brief discussion of the technique and theory of optimal ordered dither pattern design is necessary.

II. Optimal Pattern Design For Conventional Tilings

Optimal pattern design involves placing the dots, and hence the thresholds, within the halftone cell in such a manner that the resulting bi-level pattern faithfully reproduces the desired gray level while simultaneously inhibiting the disturbing appearance of patterns and textures. Although it is not at all obvious how this might be accomplished, the Fourier domain provides a convenient arena in which to approach this problem.

By way of example, consider the representation of a gray level by the periodic arrangement of $N \times N$ dither cells as shown in Figure 1a. This rectangular tiling is the conventional form of image plane tessellation used in digital halftoning.

If $f_{x,y}$ is the absorptance of the dithered image at point (x,y), then $f_{x,y} = 0, 1$. Since the dither cells are identical, clearly $f_{x,y} = f_{x+kN,y+lN} \quad \forall k,l$. This periodic two-dimensional sequence can be represented by a Fourier series

$$f_{x,y} = \sum_{u=-N}^{N} \sum_{v=-N}^{N} F_{u,v} e^{j2\pi(\frac{ux+vy}{N})} \tag{3}$$

where the coefficients $F_{u,v}$ are given by

$$F_{u,v} = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_{x,y} e^{-j2\pi(\frac{ux+vy}{N})} \tag{4}.$$

Figure 2a shows the locations of the baseband spectral components; each dot indicating the presence of a non-zero spectral component.

Vision researchers have shown conclusively that when a bi-level dot pattern attempts to simulate a uniform shade of gray, it's low frequency components make the greatest contribution to the overall visibility of patterns and granular textures [5]. Thus optimal dot patterns are those designed in a fashion that minimizes their low frequency components. With this in mind, Bayer[4] designed an optimal dot pattern sequence for a rectangularly tessellated image plane. The following paragraphs briefly summarize the technique used by Bayer to create an optimal dot pattern.

Referring to Figure 2a, it is seen that the spectral components nearest the origin are $F_{0,1}, F_{1,0}, F_{0,-1}$ and $F_{-1,0}$. An acceptable dot pattern, therefore minimizes or suppresses these components as often as possible throughout the entire range of gray levels. To design such a dot pattern assume that with the addition of the $k'th$ dot $F_{0,1} = F_{1,0} = F_{0,-1} = F_{-1,0} = 0$. It can be seen that the addition of the $k + 1'st$ dot at position (x_α, y_α) introduces a non-zero term to (4). Hence, the $k+1'st$ gray level exhibits a higher degree of pattern visibility than does the $k'th$ gray level. If, however, the $k+2'nd$ dot is placed at an "optimal" position, the non-zero term introduced by the $k+1'st$ dot can be counterbalanced and once again $F_{01} = F_{10} = F_{0-1} = F_{-10} = 0$. Denote this optimal position (x_β, y_β). It can be shown that the construction rule for the placement of successive pairs of dots is $|x_\beta - x_\alpha| = \frac{N}{2}$, $|y_\beta - y_\alpha| = \frac{N}{2}$. Repeated application of this construction rule guarantees that half of the binary dot patterns are void of the $F_{0,1}, F_{1,0}, F_{0,-1}$ and $F_{-1,0}$ spectral components and hence to a significant degree render their gray levels without the appearance of false patterns.

The dither pattern can be further improved by noting that although the relative positions of two successive dots are fixed by the first construction rule the pattern has not yet been uniquely determined. Indeed, several different patterns can be created using the first construction rule with some being significantly better than others. To create the optimal pattern (ie. one which minimizes the most low frequency components over the entire gray scale) a second construction rule guaranteeing suppression of the four next lowest spectral components, in this case $F_{1,1}, F_{1,1}, F_{1,-1}$ and $F_{-1,1}$ must be derived and applied to successive sets of four dots. Further construction rules can be derived and applied as needed. Successful recursive application of the construction rules results in a dither pattern in which every second gray level is void of four spectral components, every fourth gray level, eight spectral components, every eighth gray level, twelve components and so on. Table I summarizes the construction rules derived by Bayer[4] for the above example. The resulting dither pattern known as the "grating dot profile", is shown in Figure 3a for $N = 4$.

It is seen, therefore, that the technique of creating a good dither pattern is a three part process consisting of identifying and ranking non-zero spectral components in order of suppression preference, deriving the respective construction rules and finally recursively applying these rules to create the dither pattern.

The previous example detailed the derivation of a halftone pattern which optimally suppressed on-axis and on-diagonal spectral components arising out of a rectangular tessellation of the image plane. An inspection of the grating dot profile's halftone gray scale shown in Figure 4a reveals certain inadequacies which were documented by Allebach [6]. With each successive gray level, the fundamental spectral component shifts back and forth between the $(\pm1, 0), (0, \pm1)$ and $(\pm1, \pm1)$ orders. Since the eye is more sensitive to vertically and horizontally oriented patterns as well as longer wavelengths, the effect of this particular dot pattern is to produce an undesirable distinctness between adjacent gray levels. Furthermore, it can be seen that gray levels with fundamental components located at the $(\pm1, 0)$ and $(0, \pm1)$ orders show a high degree of pattern visibility. Based on the limitations of the "optimal dither cell", it is clear that the ordered dither algorithm requires some modification before halftone image quality can be improved further.

III. Vector Dithering

To see how this may be accomplished, consider the $MN \times MN$ dither cell which contains M^2 complete cycles of the fundamental $N \times N$ dither cell. It is easily seen that the addition of a single dot within the $N \times N$ cell results in the simultaneous placement of a vector of M^2 dots within the $MN \times MN$ cell whose positions relative to one another is governed only by the size of the fundamental cell and the tessellation. Note that increasing the dither cell size M-fold results in the formation of new spectra lying between the spectral components of the $N \times N$ cell. Hence, the $(\pm 1, 0)$ and $(0, \pm 1)$ orders for the fundamental $N \times N$ cell correspond to the $(\pm M, 0)$ and $(0, \pm M)$ orders for the larger $MN \times MN$ cell. For the standard rectangular tessellation, the highly structured vector of dots within the $MN \times MN$ cell guarantees suppression of all newly formed spectra for all gray levels. However, the original fundamental screen spectral components, $(\pm M, 0)$,$(0, \pm M)$,etc remain uninhibited. Thus, although the grating dot profile assures maximum possible suppression for on-axis and on-diagonal components, the rectangular tessellation of the image plane puts an inherent limitation on the grating dot profile's performance by ensuring that the fundamental component oscillates between the $(\pm M, 0), (0, \pm M)$ and $(\pm M, \pm M)$ orders for any M.

If, on the other hand, the M^2 dots are arranged in an alternate fashion, the pattern of spectral suppression changes, leading to two apparent benefits. The first benefit is that it becomes possible to design a dither pattern whose fundamental frequency lies beyond the $(\pm M, 0)$ and $(0, \pm M)$ orders resulting in reduced visibility of false textures and contours. The second immediate benefit is the creation of a screen whose fundamental spectral component lies in a region of the u, v-spectral plane that the human eye is less sensitive to (ie. 45°) Thus as the fundamental spectral component oscillates between orders, the change is less perceivable to the human visual system resulting in a smoother gray level transition. Since the configuration of M^2 dots is implicitly dependent upon the image plane tessellation, changing the tiling seems an appropriate solution to the problem of improving the ordered dither algorithm.

For purposes of brevity, only two tessellations will be introduced and discussed in this section. The analysis and results presented for these two tilings are representative of the results attainable when unconventional tilings are used in the ordered dither algorithm.

The first tiling considered is the skewed arrangement of $3N \times N$ dither cells as shown in Figure 1b. For analysis as well as implementation simplicity this tiling can be construed as the rectangular tessellation of a $3N \times 3N$ halftone cell (also shown in Figure 1b) which contains 3 complete cycles of the fundamental $3N \times N$ halftone cell. Note that an increase in gray level in the $3N \times N$ cell results in the addition of a vector of 3 dots to the $3N \times N$ cell. Figure 2b illustrates the effect of this tessellation on the pattern of spectral suppression. Inspection of Figure 2b reveals that the unsuppressed components lie on parallel diagonal lines, in effect creating a visually acceptable angled pattern. It can also be seen that the majority of on-axis spectral components are suppressed by this tiling. Table II summarizes the construction specification for the dither pattern which best suppresses the uninhibited spectral orders and Figure 3b shows the resulting pattern for $N = 2$.

As expected, an inspection of the resulting halftone gray scale shown in Figure 4b reveals not only better tonal representations, but also smoother gray level transitions than for the grating dot profile. Despite the excellent results, it is evident from the patterns that the horizontal nature of the skewing admits a biasing. The patterns, although less noticeable, all have a similar diagonal orientation resulting because the unsuppressed spectral components lie on diagonal lines. It is clear, therefore, that we should use tessellations that suppress components in a quasi-symetric fashion.

Figure 1c reveals one such tiling which shall be referred to as the 13 point diamond tiling. Figure 2c shows the pattern of spectral suppression resulting from tessellating the image plane in this manner while Table III summarizes the construction rules for creating the dither pattern which best minimizes those components. Figure 4c shows the resulting halftone gray scale.

IV. Angled Screens

In many halftoning algorithms it becomes necessary to create a dither cell that forms a screen pattern at some desired angle to the picture scan direction. The reasons for wanting an angled screen may range from irregular picture aspect ratio to the asymetric nature of the human visual system spatial frequency response. Regardless of the reason, the use of angled screens often results in significant improvement of halftone image quality. Consequently, there has been interest in algorithms for arbitrary angled screens.

Holladay [7] introduced a convenient technique for the creation and implementation of angled screens. Summarizing the method, creating an arbitrary angled screen involves superimposing a two dimensional array of identical parallelograms on the image plane at the desired angle. Those points falling entirely inside a single parallelogram comprise the fundamental halftone cell. Figure 6 illustrates the creation of a five point 26.5° screen; the numbers indicating the arrangement of thresholds.

Although creating the physical dither cell is simple under this technique, optimally placing the thresholds with the cell is not an easy task and was not addressed by Holladay. Optimal dither patterns for screens created in this fashion can easily be constructed by invoking the concepts of vector dithering and applying them to the unconventional tilings created by the parallelogram superposition. For the above example, it can easily be shown that the 5 point 26.5° tiling can also be construed as the rectangular tessellation of a 5×5 cell which contains 5 complete cycles of original 5 point cell. Figure 7 shows the optimal threshold distribution for such a cell. It is of passing interest to note that both the 13 point diamond and 5 point 26.5° tilings result in the rectangular tesellation of the image plane by 13×13 and 5×5 magic squares respectively.

V. Conclusion

The main thrust of this paper has been to introduce a means of improving the ordered dither halftoning algorithm by the use of unconventional tilings of the image plane. It has been shown that unconventionally tiling the image plane and using the concepts of vector dithering to create an optimal angled dither pattern results in a substantial improvement in halftoning performance. Although the number of monohedral tilings is large, this paper has concentrated on a few representative tessellations that produce typical results.

It has also been shown that the vector dithering concepts can be applied to angled screens created by Holladay's technique to create optimal patterns for those types of tilings as well.

Due to space limitations, a detailed comparision of the various tessellations under an analytic performance measure has not been included. As such, it will be the primary focus of a followup paper. Areas of further research include application of vector dithering to bi-level displays of irregular aspect ratios and improved implementation algorithms.

VI. Acknowledgements

The authors gratefully acknowledge the support of the UNISYS Corporate Technology Center.

VII. References

[1] J.O. Limb, "Design of Dither Waveforms for Quantized Visual Signals," *Bell Systems Technical Journal,* 48(1969), 2555-2582.

[2] B. Lippel and M. Kurland, "The Effect of Dither On Luminance Quantization of Pictures," *IEEE Trans. Commun. Tech,* COM-19, 879-888 (1971).

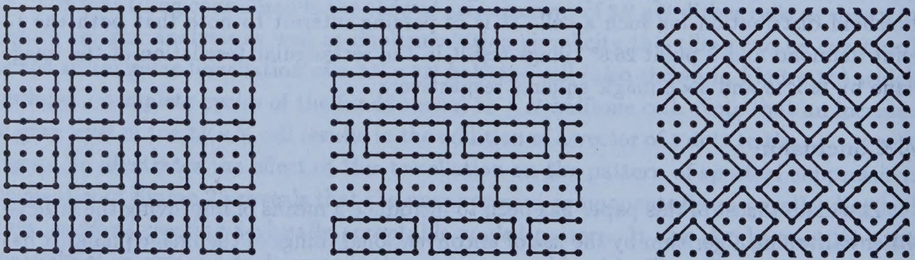
[3] C.N. Judice, J.F. Jarvis, and W.H. Ninke, "Using Ordered Dither to display Continuous-Tone Pictures on an AC Plasma Panel," *Proc. SID,* 15, 161-169 (1974)

[4] B.E. Bayer, "An Optimal Method for Two-Level Rendition of Continuous Tone Pictures," *IEEE ICC, Conference Record,* (1973),(26-11)-(26-15).

[5] F.W. Campbell and J.G. Robson, " Application of Fourier Analysis to the Visibility of Gratings," *J. Physiol,* 197, 551-566 (1968).

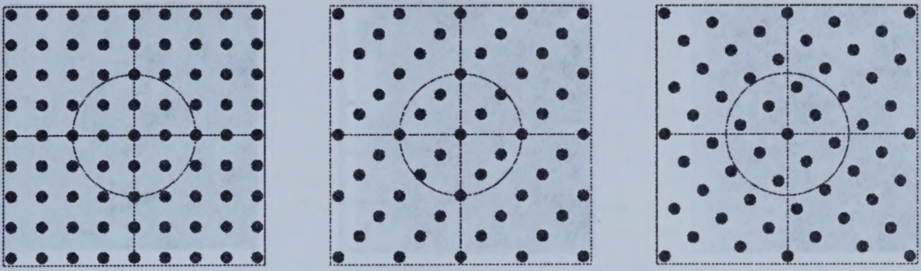
[6] J.P. Allebach and R.N. Stradling, "Computer Aided Design of Dither Signals For Binary Display of Images," *Applied Optics* , Vol 18, No. 15 (August 1979), 2708-2713.

[7] Thomas M. Holladay, "An Optimal Algorithm for Halftone Generation for Displays and Hard Copies," *Proceedings of the SID,* Vol 21, No. 2 (1980), 182-192.



(a) Standard (b) Rectangular (c) Diamond

Fig. 1 Sample Tilings



(a) Standard (b) Rectangular (c) Diamond

Fig. 2 Baseband Component Location

1	9	3	11
13	5	15	7
4	12	2	10
16	8	14	6

(a) Standard

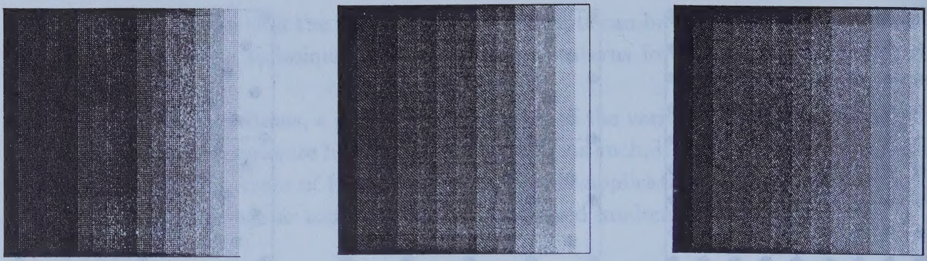
6	4	12	5	3	11
10	7	1	9	8	2
3	11	6	4	12	5
8	2	10	7	1	9
12	5	3	11	6	4
1	9	8	2	10	7

(b) Rectangular

8	3	9	13	6	1	7	12	4	10	5	11	2
4	10	5	11	2	8	3	9	13	6	1	7	12
13	6	1	7	12	4	10	5	11	2	8	3	9
11	2	8	3	9	13	6	1	7	12	4	10	5
7	12	4	10	5	11	2	8	3	9	13	6	1
3	9	13	6	1	7	12	4	10	5	11	2	8
10	5	11	2	8	3	9	13	6	1	7	12	4
6	1	7	12	4	10	5	11	2	8	3	9	13
2	8	3	9	13	6	1	7	12	4	10	5	11
12	4	10	5	11	2	8	3	9	13	6	1	7
9	13	6	1	7	12	4	10	5	11	2	8	3
5	11	2	8	3	9	13	6	1	7	12	4	10
1	7	12	4	10	5	11	2	8	3	9	13	6

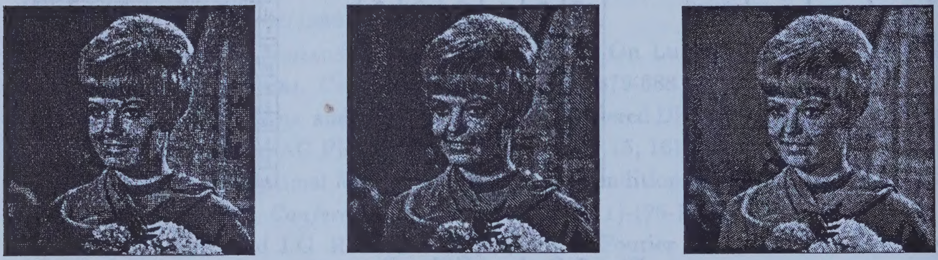
(c) Diamond

Fig. 3 Optimal Dither Patterns



(a) Standard (b) Rectangular (c) Diamond

Fig. 4 Gray Scale



(a) Standard (b) Rectangular (c) Diamond

Fig. 5 Sample Dithered Images

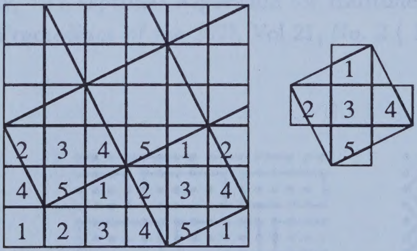

2 3 4 5 1 2
4 5 1 2 3 4
1 2 3 4 5 1
1
2 3 4
5

Fig. 6 26.5° Angled Screen Creation

5	3	1	2	4
2	4	5	3	1
3	1	2	4	5
4	5	3	1	2
1	2	4	5	3

Fig. 7 Optimal Dither Pattern 26.5° Tiling

Table I

Dither Pattern Construction Specification: *Conventional Tiling*	
Spectral Components	Construction Rule
$F_{10}, F_{01}, F_{-10}, F_{0-1}$	$\lvert x_\beta - x_\alpha\rvert = \frac{N}{2}, \lvert y_\beta - y_\alpha\rvert = \frac{N}{2}$
$F_{11}, F_{-1-1}, F_{-11}, F_{1-1}$	$\lvert x_\beta - x_\alpha\rvert = \frac{N}{2}, y_\beta = y_\alpha$
$F_{20}, F_{02}, F_{-20}, F_{0-2}$	$\lvert x_\beta - x_\alpha\rvert = \frac{N}{4}, \lvert y_\beta - y_\alpha\rvert = \frac{N}{4}$
$F_{22}, F_{-2-2}, F_{-22}, F_{2-2}$	$\lvert x_\beta - x_\alpha\rvert = \frac{N}{4}, y_\beta = y_\alpha$

Table II

Dither Pattern Construction Specification: *Rectangular Tiling*	
Spectral Components	Construction Rule
$F_{11}, F_{-1-1}, F_{-11}, F_{1-1}$	$x_\beta = x_\alpha, \lvert y_\beta - y_\alpha\rvert = \frac{3N}{2}$
$F_{2-1}, F_{1-2}, F_{-12}, F_{-21}$	$\lvert x_\beta - x_\alpha\rvert = 2N, \lvert y_\beta - y_\alpha\rvert = \frac{3N}{2}$
$F_{30}, F_{-30}, F_{0-3}, F_{03}$	$\lvert x_\beta - x_\alpha\rvert = \frac{N}{2}, \lvert y_\beta - y_\alpha\rvert = \frac{N}{2}$

Table III

Dither Pattern Construction Specification: *13 Point Diamond Tiling*	
Spectral Components	Construction Rule
$F_{32}, F_{-3-2}, F_{15}, F_{-1-5}$	$\lvert x_\beta - x_\alpha\rvert = 2, \lvert y_\beta - y_\alpha\rvert = 1$
$F_{-23}, F_{2-3}, F_{5-1}, F_{-51}$	$\lvert x_\beta - x_\alpha\rvert = 2, \lvert y_\beta - y_\alpha\rvert = 4$

LINEAR STABLE UNITY-FEEDBACK SYSTEM : NECESSARY AND SUFFICIENT CONDITIONS FOR STABILITY UNDER NONLINEAR ADDITIVE PLANT PERTURBATIONS

CHARLES A. DESOER and M. GÜNTEKİN KABULİ
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, CA 94720

Notation: (e.g. [Wil.1, Saf.1, Des.1]) Let $\mathcal{T} \subset \mathbb{R}$ and let $\mathcal{V}$ be a normed vector space. Let $\zeta := \{ F \mid F : \mathcal{T} \rightarrow \mathcal{V} \}$ be the vector space of $\mathcal{V}$–valued functions on $\mathcal{T}$. For any $T \in \mathcal{T}$, the projection map $\Pi_T : \zeta \rightarrow \zeta$ is defined by $\Pi_T F(t) := \begin{cases} F(t) & t \leq T \,,\; t \in \mathcal{T} \\ \theta_\zeta & t > T \,,\; t \in \mathcal{T} \,, \end{cases}$ where θ_ζ is the zero element in ζ. Let $\Lambda \subset \zeta$ be a normed vector space which is closed under the family of projection maps $\{ \Pi_T \}_{T \in \mathcal{T}}$. For any $F \in \Lambda$, let the norm $\| \Pi_{(\cdot)} F \| \; : \mathcal{T} \rightarrow \mathbb{R}_+$ be a nondecreasing function. The *extended space* Λ_e is defined by $\Lambda_e := \{ F \in \zeta \mid \forall T \in \mathcal{T},\; \Pi_T F \in \Lambda \}$.

A map $F : \Lambda_e \rightarrow \Lambda_e$ is said to be *causal* iff for all $T \in \mathcal{T}$, Π_T commutes with $\Pi_T F$; equivalently, $\Pi_T F = \Pi_T F \Pi_T$.

A *feedback system* is said to be *well-posed* iff for all allowed inputs, all of the signals in the system are (uniquely) determined by causal maps.

In the following we will be considering a number of function spaces closely related to Λ_e. The superscript i and the superscript o refer to "input" and "output", respectively. Let Λ_e^i and Λ_e^o be extended function spaces analogous to Λ_e except that their functions take values in the normed spaces $\mathcal{V}^i$ and $\mathcal{V}^o$, respectively; the associated projections Π_T are redefined accordingly.

A causal map $H : \Lambda_e^o \times \Lambda_e^i \rightarrow \Lambda_e$ is said to be *S-stable* iff there exists a continuous nondecreasing function $\phi_H : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that $\forall (u_1, u_2) \in \Lambda^o \times \Lambda^i$, $\| H(u_1, u_2) \| \leq \phi_H(\| u_1 \| + \| u_2 \|)$.

A well-posed (nonlinear) *feedback system* is called *S-stable* iff, for all allowed inputs, all of the signals in the feedback system are determined by causal S-stable maps.

A causal (nonlinear) map $P : \Lambda_e^i \rightarrow \Lambda_e^o$ is said to have a *right factorization* $(N_p \,, D_p \,; X_p)$ iff there exist causal S-stable maps $N_p \,, D_p$, such that

(i) $D_p : X_p \subset \Lambda_e^i \rightarrow \Lambda_e^i$ is bijective and has a causal inverse,
and (ii) $N_p : X_p \rightarrow \Lambda_e^o$, with $N_p[X_p] = P[\Lambda_e^i]$,
and (iii) $P = N_p \, D_p^{-1}$ [Vid.1, Ham.1].

X_p is called the *factorization space* of the right factorization $(N_p \,, D_p \,; X_p)$ [Ham.1].

$(N_p \,, D_p \,; X_p)$ is said to be a *normalized right-coprime factorization* of $P : \Lambda_e^i \rightarrow \Lambda_e^o$ iff
(i) $(N_p \,, D_p \,; X_p)$ is a right factorization of P,
and (ii) there exist causal S-stable maps $U_p : \Lambda_e^o \rightarrow X_p$ and $V_p : \Lambda_e^i \rightarrow X_p$ such that $U_p N_p + V_p D_p = I_{X_p}$, where I_{X_p} denotes the identity map on X_p.

1. Assumption : Consider the well-posed *linear* unity-feedback system $S(P, C)$ in Figure 1 : the plant and the compensator are given by causal *linear* (not necessarily time-invariant) maps $P : \Lambda_e^i \rightarrow \Lambda_e^o$ and $C : \Lambda_e^o \rightarrow \Lambda_e^i$ which have *normalized* right-coprime factorizations $(N_{pr}, D_{pr}; \Lambda_e^i)$ and $(N_{cr}, D_{cr}; \Lambda_e^o)$, respectively; N_{pr}, D_{pr}, N_{cr} and D_{cr} are *linear* maps (see for example [Man.1] for the continuous-time linear time-varying case).

2. Lemma : Let Assumption 1 hold. From Figure 1, we obtain the causal S-stable linear map M, defined by

$$M : \Lambda_e^o \times \Lambda_e^i \rightarrow \Lambda_e^o \times \Lambda_e^i \;,\; M : \begin{bmatrix} \xi_c^0 \\ \xi_p^0 \end{bmatrix} \mapsto \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} D_{cr} & N_{pr} \\ -N_{cr} & D_{pr} \end{bmatrix} \begin{bmatrix} \xi_c^0 \\ \xi_p^0 \end{bmatrix} . \tag{1}$$

Then the system $S(P, C)$ is S-stable if and only if the bijective map M in (1) has a causal S-stable inverse.

□

Let Assumption 1 hold and let $S(P, C)$ be S-stable. Then the map M defined in (1) has a causal S-stable inverse M^{-1}. This inverse map is linear and is given by

$$M^{-1} : \Lambda_e^o \times \Lambda_e^i \rightarrow \Lambda_e^o \times \Lambda_e^i \;,\; M^{-1} : \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \mapsto \begin{bmatrix} \xi_c^0 \\ \xi_p^0 \end{bmatrix} = \begin{bmatrix} D_{pl} & -N_{pl} \\ N_{cl} & D_{cl} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} , \tag{2}$$

where N_{cl}, D_{cl}, N_{pl} and D_{pl} denote the four submaps of M^{-1}; they are causal *linear* S-stable maps.

Let $\Delta P : \Lambda_e^i \rightarrow \Lambda_e^o$ ($\Delta P : \Lambda_e^o \rightarrow \Lambda_e^i$) be any causal *nonlinear* map such that the feedback system $S((P, \Delta P)_{22}, C)$ in Figure 2 is well-posed.

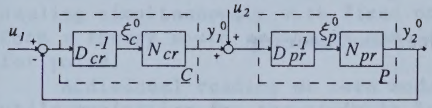


Figure 1 The feedback system $S(P, C)$

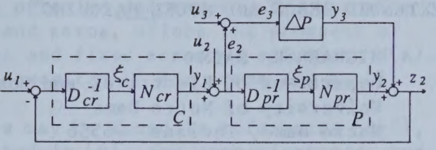


Figure 2 The feedback system $S((P, \Delta P)_{22}, C)$

We consider four perturbation cases :

i) $S((P, \Delta P)_{22}, C)$: This perturbed system is obtained from $S(P, C)$ by replacing P with a nonlinear perturbed version $(P, \Delta P)_{22}$ which has *two* inputs (e_2, u_3) and *two* observed outputs (y_2, y_3).

Suppose that in case i) we observe only z_2; then we obtain

ii) $S((P, \Delta P)_{21}, C)$: the perturbation $(P, \Delta P)_{21}$ has *two* inputs (e_2, u_3) and *one* observed output z_2.

Suppose that we set $u_3 \equiv 0$ in case i), then we obtain

iii) $S((P, \Delta P)_{12}, C)$: the perturbation $(P, \Delta P)_{12}$ has *one* input e_2 and *two* observed outputs (y_2, y_3).

Suppose that in case i), we set $u_3 \equiv 0$ and observe only z_2; then we obtain

iv) $S(P + \Delta P, C)$: the perturbation $P + \Delta P$ has *one* input e_2 and *one* observed output z_2.

Note that for $i, j = 1, 2$, the $(i+1)$-input system $S((P, \Delta P)_{ij}, C)$ is S-stable iff the $j+1$ outputs (i.e. j outputs of $(P, \Delta P)_{ij}$ and y_1) are determined by causal S-stable maps.

3. Theorem : (Necessary and Sufficient Condition for Robustness) Let Assumption 1 hold. Let the *linear* system $S(P, C)$ be S-stable. Then for any causal *nonlinear* map $\Delta P : \Lambda_e^i \to \Lambda_e^o$,

i) the well-posed $S((P, \Delta P)_{22}, C)$ is S-stable if and only if $\Delta P (I + N_{cr} D_{pl} \Delta P)^{-1}$ is S-stable.

ii) the well-posed $S((P, \Delta P)_{21}, C)$ is S-stable if and only if $D_{pl} \Delta P (I + N_{cr} D_{pl} \Delta P)^{-1}$ is S-stable.

iii) the well-posed $S((P, \Delta P)_{12}, C)$ is S-stable if and only if $\Delta P (I + N_{cr} D_{pl} \Delta P)^{-1} D_{pr}$ is S-stable.

iv) the well-posed $S(P + \Delta P, C)$ is S-stable if and only if $D_{pl} \Delta P (I + N_{cr} D_{pl} \Delta P)^{-1} D_{pr}$ is S-stable.

□

Solving for ΔP in the four necessary and sufficient conditions in Proposition 3, we obtain a characterization of the set of all nonlinear perturbations ΔP for which the perturbed system remains S-stable (called addmissable perturbations) :

4. Corollary : (Characterization of admissable ΔP 's) Let Assumption 1 hold. Let the linear system $S(P, C)$ be S-stable. Then

i) the well-posed $S((P, \Delta P)_{22}, C)$ is S-stable if and only if $\Delta P = Q (I - N_{cr} D_{pl} Q)^{-1}$ for some causal S-stable map Q.

ii) the well-posed $S((P, \Delta P)_{21}, C)$ is S-stable if and only if $\Delta P = D_{pl}^{-1} Q (I - N_{cr} Q)^{-1}$ for some causal S-stable map Q.

iii) the well-posed $S((P, \Delta P)_{12}, C)$ is S-stable if and only if $\Delta P = Q (I - N_{cl} Q)^{-1} D_{pr}^{-1}$ for some causal S-stable map Q.

iv) the well-posed $S(P + \Delta P, C)$ is S-stable if and only if $\Delta P = D_{pl}^{-1} Q (D_{pr} - N_{cr} Q)^{-1}$ for some causal S-stable Q.

□

In the case that the plant P has right- and left-coprime factorizations (see [Vid.2] for the linear time-invariant case, [Man.1] for the continuous-time linear time-varying case), the set of all stabilizing compensators C for the nominal plant is given in terms of the plant factorizations and a free linear stable parameter; hence the factors N_{cr} and N_{cl} of C in Corollary 4 would also depend on this free linear stable parameter.

References

[Des.1] C. A. Desoer, M. Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press, 1975.

[Ham.1] J. Hammer, "*Nonlinear Systems, Stabilization and Coprimeness*", Int. J. of Control, vol. 42 no. 1, pp. 1-20, 1985.

[Man.1] V. Manousiouthakis, "*On Time-Varying Control*", Proc. ACC Minneapolis, pp. 2050-2053, 1987.

[Saf.1] M. G. Safonov, *Stability and Robustness of Multivariable Feedback Systems*, MIT Press, 1980.

[Vid.1] M. Vidyasagar, N. Viswanadham, "*Stabilization of Linear and Nonlinear Dynamical Systems using an Observer-Controller Configuration*", Syst. & Contr. Lett., vol. 1 no. 2 , pp. 87-91, 1981.

[Vid.2] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*, MIT Press, 1985.

[Wil.1] J. C. Willems, *The Analysis of Feedback Systems*, MIT Press, 1971.

EXTENDED ZEROS AND MODEL MATCHING*

MICHAEL K. SAIN
Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, Indiana 46556

BOSTWICK F. WYMAN
Department of Mathematics
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

In a recent study, Sain, Peczkowski, and Wyman [1] have introduced a module Z_Ω of fixed, extended zeros in solutions $M(z)$ to the model matching equation $T(z) = P(z)M(z)$. This investigation examines Z_Ω further. First, Z_Ω must be a factor module of the extended zeros of any solution. Second, Z_Ω has a torsion submodule which can be characterized in a manner similar to, but more intricate than, Z_Ω itself. Third, Z_Ω can be seen as a direct summand in a unified, pole-zero module of extended type; this approach permits a simultaneous study of poles and zeros.

1. INTRODUCTION

The recent paper [1] by Sain, Peczkowski, and Wyman has considered the question of fixed zeros in solutions $M(z)$ to the model matching equation

$$T(z) = P(z)M(z), \tag{1-1}$$

which is depicted in the commutative diagram of Figure 1. The symbols $R(z)$, $U(z)$, and $Y(z)$ denote vector spaces of finite dimension over $k(z)$, the field of transfer functions with coefficients in a field k. The symbols $T(z)$, $P(z)$, and $M(z)$ then stand for $k(z)$-linear maps, with domains and codomains as indicated in Figure 1. A principal tool in [1] was a module Z_Ω of fixed zeros associated with solutions $M(z)$ to (1-1). Z_Ω is an extended zero module of omega type, which means that it is built up from a direct summing of classical, state-space, zeros and free zeros, only recently defined.

$$\begin{tikzcd}
& R(z) \arrow[dl, dashed, "M(z)"'] \arrow[dr, "T(z)"] & \\
U(z) \arrow[rr, "P(z)"] & & Y(z)
\end{tikzcd}$$

Figure 1

This paper continues the investigation begun in [1], by examining Z_Ω further. First, we will show that Z_Ω must be a factor module of $Z_\Omega(M(z))$, where the notation means the extended zero module of omega type attached to any solution $M(z)$. Second, we shall develop a description of the state-space-type submodule of Z_Ω, in terms of short exact sequences. This description turns out to be more involved than that given previously [1] for Z_Ω. Third, we define a unified, pole-zero module of omega type, and

---

*The work of the first author was supported by The Ohio State University's Distinguished Visiting Professorship, by the University of Notre Dame's Frank M. Freimann Chair in Electrical Engineering, and by the National Science Foundation under Grant ECS-84-05714.

introduce a corresponding fixed module of poles and zeros. The use of a unified module, containing both poles and zeros, offers the prospect of dealing simultaneously with fixed poles and fixed zeros of solutions $M(z)$. Such a theory would reduce in special cases to [1] for zeros and to [2] for poles.

Additional reading on zero modules may be found in references [3-5], while motivation for the study is located in [6], for communications, and in [7], for control.

2. POLE AND ZERO MODULES

Let $k[z]$ be the ring of polynomials in z with coefficients from a field k; and let $k(z)$ be the induced quotient field. Suppose that R, U, and Y are vector spaces of finite dimension over k. Then because $k[z]$ and $k(z)$ are k-vector spaces, we can employ the k-bilinear tensor product to construct modules

$$R[z] = k[z] \otimes_k R, \quad U[z] = k[z] \otimes_k U, \quad Y[z] = k[z] \otimes_k Y, \tag{2-1}$$

over the ring $k[z]$ and vector spaces

$$R(z) = k(z) \otimes_k R, \quad U(z) = k(z) \otimes_k U, \quad Y(z) = k(z) \otimes_k Y, \tag{2-2}$$

over the field $k(z)$. In this section, we discuss the module theoretic character of $k[z]$-zeros and poles for $k(z)$-linear maps, such as $M(z) : R(z) \to U(z)$. Particular attention is paid to the Ω-extended zeros of such maps, as introduced by Sain, Peczkowski, and Wyman [1] in 1987.

The pole module of a $k(z)$-linear map, such as $M(z)$, is defined to be the $k[z]$-factor module

$$P\big(M(z)\big) = MR[z]/\{U[z] \cap MR[z]\}. \tag{2-3}$$

Module (2-3) is finitely generated and torsion. As a k-vector space, $P\big(M(z)\big) = X_P$, the customary state space of a minimal realization of $M(z)$. Moreover, scalar multiplication by z, in the $k[z]$-module $P\big(M(z)\big)$, defines a k-linear map $A_P : X_P \to X_P$; and we call A_P the pole-state dynamics matrix. For zeros, an analogous module has been introduced by Wyman and Sain [3] in 1981. This module, called the zero module of $M(z)$, is defined to be the $k[z]$-factor module

$$Z\big(M(z)\big) = \{U[z] \cap MR(z)\}/\{U[z] \cap MR[z]\}. \tag{2-4}$$

Module (2-4) is also finitely generated and torsion, and so has a state-space interpretation of its own. As a k-vector space, $Z\big(M(z)\big) = X_Z$, which is of finite dimension over k. Once again, scalar multiplication by z in the $k[z]$-module brings about a k-linear map on X_Z into itself, which we denote by A_Z. The space X_Z is not the usual state space. To avoid confusion, we refer to X_P as the pole-state space and to X_Z as the zero-state space. A_Z is then the zero-state dynamics matrix. Note that zero-state, as used in this terminology, does not refer to initial conditions. Instead, we refer to two distinct state spaces attached to $M(z)$, one for poles and the other for zeros.

The modules $P\big(M(z)\big)$ and $Z\big(M(z)\big)$ are of the ordinary state-space type. In an examination of the fixed zeros of solutions $M(z)$ to (1-1), however, it has been shown [1] that certain types of extended zeros enter the picture. We call these extended zeros of omega type, or simply Ω-zeros. The Ω-zero module of a $k(z)$-linear map, such as $M(z) : R(z) \to U(z)$, is defined to be the $k[z]$-factor module

$$Z_\Omega\big(M(z)\big) = U[z]/\{U[z] \cap MR[z]\}. \tag{2-5}$$

Module (2-5) is finitely generated; but it is not a torsion module. From an examination of (2-4) and (2-5), we see that $Z\big(M(z)\big)$ is a submodule of $Z_\Omega\big(M(z)\big)$. Thus there is a natural inclusion

$$i : Z\big(M(z)\big) \to Z_\Omega\big(M(z)\big) \tag{2-6}$$

having cokernel isomorphic as a $k[z]$-module to

$$\Omega\big(M(z)\big) = U[z]/\{U[z] \cap MR(z)\}. \tag{2-7}$$

Now $\Omega\big(M(z)\big)$ is finitely generated; and $k[z]$ is a principal ideal domain. We wish to determine whether or not $\Omega\big(M(z)\big)$ has any nonzero torsion elements. Suppose that it does, and represent such an element by $u(z)$ in $U[z]$. For such a $u(z)$, there must exist a nonzero polynomial $p(z)$ in $k[z]$ such that

$$p(z)u(z) = M(z)r(z) \tag{2-8a}$$

for an appropriate vector $r(z)$ in $R(z)$. But then

$$u(z) = M(z)\big[r(z)/p(z)\big], \tag{2-8b}$$

so that $u(z)$ is in $M(z)R(z)$ and must be equivalent to zero in (2-7). Thus $\Omega\big(M(z)\big)$ has no nonzero torsion elements. Taking into account its finite generation, together with the nature of $k[z]$, we then have that $\Omega\big(M(z)\big)$ is a free module. Next, consider that the inclusion

$$U[z] \cap MR[z] \subset U[z] \cap MR(z) \tag{2-9}$$

guarantees the existence of an epic, $k[z]$-linear map from $Z_\Omega\big(M(z)\big)$ onto $\Omega\big(M(z)\big)$. It follows that the Ω-zero module is isomorphic to a direct sum

$$Z_\Omega\big(M(z)\big) \approx \Omega\big(M(z)\big) \oplus Z\big(M(z)\big) \tag{2-10}$$

of $k[z]$-zeros in $\Omega\big(M(z)\big)$ with $k[z]$-zeros in $Z\big(M(z)\big)$. We shall refer to the module (2-7) as the free zero module of $M(z)$. Ω-zeros, then, are seen as being comprised of free zeros and zeros of state-space type, in accordance with an exact sequence

$$0 \to Z\big(M(z)\big) \to Z_\Omega\big(M(z)\big) \to \Omega\big(M(z)\big) \to 0. \tag{2-11}$$

Free zeros of a $k(z)$-linear map arise when the cokernel of that map fails to be zero. This may be seen from (2-7), where the entire module $U[z]$ becomes equivalent to zero when $\operatorname{im} M(z) = U(z)$. Unlike zeros of state-space type, free zeros are not detected by the usual rank reduction tests. Nonetheless, they fit together naturally in a study of fixed zeros in model matching problems.

3. FIXED Ω-ZEROS

The idea of fixed zeros for solutions $M(z)$ to the model matching equation (1-1) can be motivated quickly and directly from an examination of the single-input, single-output, case in which the dimensions of R, U, and Y are each equal to unity. Suppose that

$$t(z) = p(z)m(z), \tag{3-1}$$

where all entries are in $k(z)$, and write the given fractions in relatively prime form

$$t(z) = n_t(z)/d_t(z), \quad p(z) = n_p(z)/d_p(z). \tag{3-2}$$

It follows that

$$m(z) = [n_t(z)d_p(z)]/[d_t(z)n_p(z)], \tag{3-3}$$

from which we see that the zeros of $m(z)$ are the zeros of $t(z)$ which are not zeros of $p(z)$, together with the poles of $p(z)$ which are not poles of $t(z)$. This idea is direct and intuitive; but it does not extend to the multivariable case without certain conceptual adjustments. In particular, the natural generalization of (3-1) to (3-3) makes use of Ω-zeros.

For the study of fixed zeros in solutions of (1-1), Sain, Peczkowski, and Wyman [1] have introduced a module

$$Z_\Omega = \frac{PU[z] + TR[z]}{TR[z]} \approx \frac{PU[z]}{PU[z] \cap TR[z]} \tag{3-4}$$

of fixed zeros. The module Z_Ω is finitely generated; but it is not a torsion module. Consider the torsion submodule of Z_Ω. If $y(z)$ is a torsion element representative, then there exists a nonzero polynomial $p(z)$ such that

$$p(z)y(z) = T(z)r(z), \quad r(z) \in R[z]. \tag{3-5}$$

But (3-5) implies that $y(z) \in T(z)R(z)$. Conversely, if $y(z) \in T(z)R(z)$, then (3-5) can be satisfied for appropriate choices of $p(z)$ and $r(z)$. So the torsion submodule of Z_Ω is given by

$$Z'_\Omega = \frac{PU[z] \cap TR(z) + TR[z]}{TR[z]} \approx \frac{PU[z] \cap TR(z)}{PU[z] \cap TR[z]}. \tag{3-6}$$

Using a natural inclusion, we can insert Z'_Ω into Z_Ω and produce a cokernel which is isomorphic to

$$\Omega = \frac{PU[z] + TR[z]}{PU[z] \cap TR(z) + TR[z]} \approx \frac{PU[z]}{PU[z] \cap TR(z)}. \tag{3-7}$$

Now Ω is a torsion-free module. To see this, suppose that $y(z)$ represents a torsion element; then

$$p(z)y(z) = T(z)r(z), \quad r(z) \in R(z), \tag{3-8}$$

for some nonzero polynomial $p(z)$. Consequently, there is a polynomial $q(z)$ such that

$$\big(q(z)p(z)\big)y(z) = T(z)\big(q(z)r(z)\big), \quad q(z)r(z) \in R[z]. \tag{3-9}$$

But this means that $y(z)$ is equivalent to zero in Z_Ω as desired. As a finitely generated module over a principal ideal domain, Ω is also free. We thus have a short exact sequence

$$0 \to Z'_\Omega \to Z_\Omega \to \Omega \to 0; \tag{3-10}$$

and, because Ω is free, and a factor module of Z_Ω, we can also write

$$Z_\Omega \approx \Omega \oplus Z'_\Omega. \tag{3-11}$$

Our next goal is to study the role of Z_Ω in solutions to (1-1).

Theorem 1. Let $\operatorname{im} T(z) \subset \operatorname{im} P(z)$; and suppose that $M(z)$ is a solution to the equation $T(z) = P(z)M(z)$. Then there exists an epic, $k[z]$-linear map

$$\beta_\Omega(z) : Z_\Omega\bigl(M(z)\bigr) \to Z_\Omega, \tag{3-12}$$

so that Z_Ω is a factor module of the Ω-zero module of $M(z)$.

Discussion of Theorem: Because the fixed module Z_Ω depends only upon $T(z)$ and $P(z)$, and not upon the solution $M(z)$, we see that it describes an essential character of Ω-zeros for all solutions to the model matching problem.

Proof of Theorem: From the commutative diagram

$$\begin{tikzcd}
U[z] \cap MR[z] \arrow[r] \arrow[d, "P(z)"'] & U[z] \arrow[d, "P(z)"] \\
PU[z] \cap TR[z] \arrow[r] & PU[z]
\end{tikzcd}$$

Figure 2

of Figure 2, which is made up of natural inclusions and restrictions of $P(z)$, we can induce the map $\beta_\Omega(z)$, which is epic because the restriction of column two is epic.

We remark that the kernel of $\beta_\Omega(z)$ is given by

$$U[z] \cap \{MR[z] + \ker P(z)\}/U[z] \cap MR[z]. \tag{3-13}$$

Torsion elements of (3-13) are associated with

$$\ker P(z) \cap MR(z); \tag{3-14}$$

and these can always be removed by solution design. In that case, (3-13) becomes a free module. More generally, we can form the torsion submodule

$$U[z] \cap \{MR[z] + \ker P(z) \cap MR(z)\}/U[z] \cap MR[z] \tag{3-15}$$

and break (3-13) into a direct sum of (3-15) and the free module

$$U[z] \cap \{MR[z] + \ker P(z)\}/U[z] \cap \{MR[z] + \ker P(z) \cap MR(z)\}. \tag{3-16}$$

4. DESCRIPTION OF Z'_Ω

In Section 3, we showed that the fixed module Z_Ω, defined in (3-4), appears as a factor module in every solution $M(z)$ to the model matching equation (1-1). Moreover, it was shown that Z_Ω has a torsion submodule Z'_Ω as given in (3-6). From these two results, it is natural to inquire about the nature of Z'_Ω.

We begin with the commutative diagram of Figure 3. In this diagram, both rows and columns are natural inclusions. Notice, in the figure, that the second column, when extended to a short exact sequence, has its factor module equal to Z'_Ω. Observe also that the first row extends into

$$\begin{tikzcd}
Y[z] \cap TR[z] \arrow[r] \arrow[d] & TR[z] \arrow[d] \\
Y[z] \cap \{PU[z] \cap TR(z) + TR[z]\} \arrow[r] & PU[z] \cap TR(z) + TR[z]
\end{tikzcd}$$

Figure 3

a short exact sequence with factor module $P(T(z))$. From Figure 3, we find a monic, $k[z]$-linear map from $P(T(z))$ into the module

$$ {}_TX_P = \frac{PU[z] \cap TR(z) + TR[z]}{Y[z] \cap \{PU[z] \cap TR(z) + TR[z]\}} . \tag{4-1}$$

The cokernel of this map is denoted by X'_z. It is possible to relate the module (4-1) to the pole module of the composite map $[T(z) \quad P(z)] : R(z) \oplus U(z) \to Y(z)$. The basic idea is shown in the commutative diagram of Figure 4,

$$\begin{tikzcd}
Y[z] \cap \{PU[z] \cap TR(z) + TR[z]\} \arrow[r] \arrow[d] & PU[z] \cap TR(z) + TR[z] \arrow[d] \\
Y[z] \cap \{PU[z] + TR[z]\} \arrow[r] & PU[z] + TR[z]
\end{tikzcd}$$

Figure 4

where rows and columns are again natural inclusions. From this diagram, we induce a monic, $k[z]$-linear map on ${}_TX_P$ into $P([T(z) \quad P(z)])$, with cokernel isomorphic to

$$ X_k = \frac{PU[z] + TR[z]}{PU[z] \cap TR(z) + TR[z] + Y[z] \cap \{PU[z] + TR[z]\}} . \tag{4-2}$$

Returning now to Figure 3, we define the factor module induced by column one to be Z'_z, and point out that it is a submodule of $Z(T(z))$, by the inclusion

$$ PU[z] \cap TR(z) + TR[z] \subset TR(z). \tag{4-3}$$

A natural inclusion of Z'_z into $Z(T(z))$ induces a factor module isomorphic to

$$ {}_TZ_P = \frac{Y[z] \cap TR(z)}{Y[z] \cap \{PU[z] \cap TR(z) + TR[z]\}} . \tag{4-4}$$

Module (4-4) can be related to the zero module $Z([T(z) \quad P(z)])$. Refer to the commutative diagram of Figure 5

$$\begin{tikzcd}
Y[z] \cap \{PU[z] \cap TR(z) + TR[z]\} \arrow[r] \arrow[d] & Y[z] \cap TR(z) \arrow[d] \\
Y[z] \cap \{TR[z] + PU[z]\} \arrow[r] & Y[z] \cap \{TR(z) + PU(z)\} = Y[z] \cap PU(z)
\end{tikzcd}$$

Figure 5

of natural inclusions. This diagram induces a monic, $k[z]$-linear map on ${}_TZ_P$ into $Z([T(z) \quad P(z)])$, with cokernel isomorphic to

$$Z_k = \frac{Y[z] \cap PU(z)}{Y[z] \cap TR(z) + Y[z] \cap \big(TR[z] + PU[z]\big)} . \tag{4-5}$$

We summarize these results in a second theorem.

Theorem 2. Let $\operatorname{im} T(z) \subset \operatorname{im} P(z)$; and let Z'_Ω be the torsion submodule of the fixed module Z_Ω of Ω-zeros in (3-4). Then there exist $k[z]$-modules ${}_TZp$, Z_k, ${}_TXp$, X_k, Z'_z, and X'_z, together with appropriate $k[z]$-linear maps, such that the following five short sequences are exact:

$$0 \to {}_TXp \to P\big(\big[T(z) \quad P(z)\big]\big) \to X_k \to 0; \tag{4-6a}$$

$$0 \to P\big(T(z)\big) \to {}_TXp \to X'_z \to 0; \tag{4-6b}$$

$$0 \to {}_TZp \to Z\big(\big[T(z) \quad P(z)\big]\big) \to Z_k \to 0; \tag{4-6c}$$

$$0 \to Z'_z \to Z\big(T(z)\big) \to {}_TZp \to 0; \tag{4-6d}$$

$$0 \to Z'_z \to Z'_\Omega \to X'_z \to 0. \tag{4-6e}$$

Proof: Sequences (4-6a) - (4-6d) have been established in the prelude to the theorem. Sequence (4-6e) is a consequence of the requirements on cokernels of columns in Figure 3, including the induced column.

Remark: The study of Z'_Ω then begins with the finitely generated, torsion poles and zeros of $T(z)$ and of $\big[T(z) \quad P(z)\big]$. From the submodules ${}_TZp$ of $Z\big(\big[T(z) \quad P(z)\big]\big)$ and ${}_TXp$ of $P\big(\big[T(z) \quad P(z)\big]\big)$, together with $Z\big(T(z)\big)$ and $P\big(T(z)\big)$, we can find Z'_z and X'_z, the building blocks of Z'_Ω. Notice that, if $\operatorname{im} T(z) = \operatorname{im} P(z)$, both Z_k and X_k vanish. In that case, we obtain the sequences

$$0 \to P\big(T(z)\big) \to P\big(\big[T(z) \quad P(z)\big]\big) \to X'_z \to 0; \tag{4-7a}$$

$$0 \to Z'_z \to Z\big(T(z)\big) \to Z\big(\big[T(z) \quad P(z)\big]\big) \to 0; \tag{4-7b}$$

$$0 \to Z'_z \to Z'_\Omega \to X'_z \to 0; \tag{4-7c}$$

and this result then reduces to [1, Theorem 2], because Ω is zero.

It has been shown in [1, Theorem 2] that Z_Ω satisfies three short exact sequences

$$0 \to P\big(T(z)\big) \to P\big(\big[T(z) \quad P(z)\big]\big) \to X_p \to 0; \tag{4-8a}$$

$$0 \to Z_p \to Z_\Omega\big(T(z)\big) \to Z_\Omega\big(\big[T(z) \quad P(z)\big]\big) \to 0; \tag{4-8b}$$

$$0 \to Z_p \to Z_\Omega \to X_p \to 0; \tag{4-8c}$$

and so we see that the description of Z'_Ω is somewhat more complicated. In a certain sense, then, Ω-zeros appear to be more naturally suited to model matching studies than zeros of classical, state-space type.

5. THE FIXED Ω-POLE-ZERO MODULE

In this section, we show how to generalize the results of Sections 3-4 so as to include at one and the same time both poles and zeros. This offers a single theory which is comprehensive enough to include both fixed zeros and fixed poles, the latter having been formerly studied by Conte, Perdon, and Wyman [2]. The vehicle for this purpose is a unified pole-zero module of omega type.

Consider the $k[z]$-factor module

$$PZ_\Omega\big(M(z)\big) = \{U[z] + MR[z]\}/\{U[z] \cap MR[z]\}. \qquad (5\text{-}1)$$

Module (5-1) is finitely generated, but may contain non-torsion elements. Notice that both $P\big(M(z)\big)$ and $Z_\Omega\big(M(z)\big)$ are submodules of (5-1). Clearly,

$$P\big(M(z)\big) + Z_\Omega\big(M(z)\big) = PZ_\Omega\big(M(z)\big), \qquad (5\text{-}2a)$$

$$P\big(M(z)\big) \cap Z_\Omega\big(M(z)\big) = 0. \qquad (5\text{-}2b)$$

Consequently, $PZ_\Omega\big(M(z)\big) = P\big(M(z)\big) \oplus Z_\Omega\big(M(z)\big)$. In view of this direct sum, we shall refer to (5-1) as the unified, pole-zero module of omega type for $M(z)$, or simply as the Ω-pole-zero module of $M(z)$.

The idea of PZ modules may be used to develop fixed modules of poles and zeros for the model matching problem.

Theorem 3. Let $\operatorname{im} T(z) \subset \operatorname{im} P(z)$; and suppose that $M(z)$ is a solution to the equation $T(z) = P(z)M(z)$. Define a module

$$PZ_\Omega = \{PU[z] + TR[z]\}/\{PU[z] \cap TR[z]\} \qquad (5\text{-}3)$$

of fixed poles and zeros in the solution $M(z)$. Then there exists an epic, $k[z]$-linear map

$$\tilde{\beta}_\Omega(z) : PZ_\Omega\big(M(z)\big) \to PZ_\Omega, \qquad (5\text{-}4)$$

so that PZ_Ω is a factor module of Ω-pole-zeros for every model matching solution.

Proof: The desired epimorphism is induced from the commutative diagram

$$\begin{tikzcd}
U[z] \cap MR[z] \arrow[r] \arrow[d, "P(z)"'] & U[z] + MR[z] \arrow[d, "P(z)"] \\
PU[z] \cap TR[z] \arrow[r] & PU[z] + TR[z]
\end{tikzcd}$$

Figure 6

of Figure 6, in which the rows are natural inclusions and the columns are restrictions of $P(z)$.

Discussion: The module PZ_Ω of fixed poles and zeros in the Ω-pole-zero module $PZ_\Omega\big(M(z)\big)$ of a solution to (1-1) appears as a required factor module. From the direct sum decomposition

$$PZ_\Omega = P \oplus Z_\Omega, \qquad (5\text{-}5)$$

where

$$P = TR[z]/\{PU[z] \cap TR[z]\} \qquad (5\text{-}6)$$

is the fixed pole module of [2], we see that the unified, Ω-pole-zero theory handles the fixed poles and zeros "in parallel."

The kernel of $\tilde{\beta}_\Omega(z)$ is given by

$$\frac{U[z] \cap \{MR[z] + \ker P(z)\} + \ker P(z) \cap \{U[z] + MR[z]\}}{U[z] \cap MR[z]}. \qquad (5\text{-}7)$$

From (3-13), we recognize the first term in (5-7) as the kernel of $\beta_\Omega(z)$. The second term is isomorphic to

$$\frac{\ker P(z) \cap \{U[z] + MR[z]\}}{\ker P(z) \cap U[z] \cap MR[z]} . \tag{5-8}$$

It is of interest to compare (5-8) with the kernel of the epic, k[z]-linear map from $P\big(M(z)\big)$ onto P, and induced from the commutative diagram of Figure 7:

$$\begin{tikzcd} U[z] \arrow[r] \arrow[d, "P(z)"'] & MR[z] + U[z] \arrow[d, "P(z)"] \\ PU[z] \arrow[r] & TR[z] + PU[z] \end{tikzcd}$$

Figure 7

This kernel is isomorphic to

$$\{(MR[z] + U[z]) \cap \ker P(z)\}/\{U[z] \cap \ker P(z)\}. \tag{5-9}$$

Notice that (5-9) is a factor module of (5-8). This means that (5-9) and (3-13) do not, separately, carry the information concerning kernel of $\tilde{\beta}_\Omega(z)$. In general, therefore, we may expect some interaction between poles and Ω-zeros in a design of solutions M(z).

6. EXTENSIONS

For readability, we have chosen the ring k[z] and the quotient field k(z) in which to state the results of Sections 2-5. However, all those results remain true in more general contexts, sometimes with a modest re-interpretation of physical meaning. As an example, we can replace the subring k[z] of k(z) with any subring O which contains the base field k, which has quotient field k(z), and which is a principal ideal domain. Such rings include localizations of k[z] and discrete valuation rings. An important instance of the latter is O_∞, the subring of proper transfer functions. More illustrations of rings which are useful in system theory arise from forming the intersection of discrete valuation rings. One illustration here is the ring O_{ps} of transfer functions which are proper and stable, when k = R, the real numbers.

When k[z] is replaced by O, Equations (2-1) must be changed to

$$\Omega_O R = O \otimes_k R, \tag{6-1a}$$

$$\Omega_O U = O \otimes_k U, \tag{6-1b}$$

$$\Omega_O Y = O \otimes_k Y. \tag{6-1c}$$

Elsewhere, simply replace R[z] by $\Omega_O R$, and so forth.

In fact, the ideas of the paper also extend to systems defined over more general rings, with appropriate assumptions [8].

7. CONCLUSIONS

In [1], Sain, Peczkowski, and Wyman have introduced a module Z_Ω of fixed zeros of extended, omega type for solutions to the model matching problem (1-1). It was possible to establish, in that study, that Z_Ω is comprised of a submodule of Ω-zeros of T(z) which are not Ω-zeros of [T(z) P(z)], together with a factor module of poles of [T(z) P(z)] which are not poles of T(z). The simplicity of this representation is a consequence of the use of Ω-zeros in place of zeros of the classical, state-space type. Ω-zero modules may be understood as direct sums of classical, state-space zero modules with the novel, free zero modules.

This paper carries these notions forward in three ways. First, we have shown that Z_Ω is an essential factor module of the Ω-zero module of any solution M(z) to the model matching equation (1-1). Second, we have represented and characterized the torsion submodule of Z_Ω, in terms of short exact sequences. As expected, this characterization is more intricate than that of Z_Ω in [1]. Third, we have introduced a unified, pole-zero module of omega type, together with a corresponding fixed module of poles and zeros. This fixed module is a required factor module of the unified, Ω-pole-zero module of any solution M(z).

This study further strengthens the role of Ω-zeros in system design, and points a path toward simultaneous construction of poles and zeros in solutions to model matching.

8. REFERENCES

1. M. K. Sain, J. L. Peczkowski, and B. F. Wyman, "Local Complexity in Syntheses of Nonlinear Feedback Systems," International Symposium on the Mathematical Theory of Networks and Systems, June 1987.

2. G. Conte, A. M. Perdon, and B. F. Wyman, "Fixed Poles in Transfer Function Equations," SIAM Journal on Control and Optimization, in press.

3. B. F. Wyman and M. K. Sain, "The Zero Module and Essential Inverse Systems," IEEE Transactions on Circuits and Systems, Vol. CAS-27, No. 2, pp. 112-126, February 1981.

4. B. F. Wyman and M. K. Sain, "Exact Sequences for Pole-Zero Cancellation," Proceedings International Symposium on the Mathematical Theory of Networks and Systems, pp. 278-280, August 1981.

5. G. Conte and A. M. Perdon, "Zeros of Cascade Compositions," in Frequency Domain and State Space Methods for Linear Systems, C. I. Byrnes and A. Lindquist, eds. North-Holland: Elsevier, 1986, pp. 23-34.

6. J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits," in Key Papers in the Development of Coding Theory, E. R. Berlekamp, ed. New York: IEEE Press, 1974, pp. 205-212.

7. H. H. Rosenbrock, State-Space and Multivariable Theory. New York: John Wiley and Sons, Inc., 1970.

8. G. Conte and A. M. Perdon, "An Algebraic Notion of Zeros for Systems over Rings," Proceedings International Symposium on the Mathematical Theory of Networks and Systems, 1983.

TRANSFER MATRICES, POLYNOMIAL FRACTIONS AND REALIZATION OF CONTINUOUS-TIME LINEAR TIME-VARYING SYSTEMS

EROL EMRE*[1], HENG-MING TAI**, and JIN H. SEO*

*Department of Electrical Engineering
Texas Tech University, Lubbock, Texas 79409

**Department of Electrical Engineering
University of Tulsa, Tulsa, Oklahoma 74104

ABSTRACT

An algebraic theory of transfer matrices, fractional representations and control for linear continuous-time time-varying systems based on realization theory of input-output maps is given. It is shown for the first time that the realization of linear continuous-time time-varying systems specified by an abstract input-output map (as a module homomorphism over noncommutative polynomial rings) can be established using an abstract Kalman input-output map defined over a ring of skew polynomials with time-varying coefficients. It is shown that, in fact, transfer matrices can be defined as formal power series for such systems as representation of this I-O map, and this is equivalent to the time domain input-output maps in terms of convolution. This is also shown to be an equivalent representation of a previously considered Hankel map. Using this framework, a realization theory for fractional representations is established.

We should note that although our results are developed for systems over infinite time intervals, all of them are valid over systems over finite intervals (t_0,t_1) (in fact, the latter class is known to be much easier to study).

We also show how our general results can be specialized to certain systems over nicer rings of time functions as well as to periodic systems, to yield much stronger results.

INTRODUCTION

During the past decade considerable research which unifies the state space approach (e.g. Wonham [24]) as well as the polynomial matrix approaches of Rosenbrock [21] via the algebraic approach of Kalman has been done (see e.g., our references). With these concepts and techniques in mind, a number of control problems concerning linear systems over rings have been solved by several authors (see e.g., Emre [2,3], Emre and Khargonekar [7], Khargonekar [19], and the references therein). Among them are, for instance, the study of matrix fraction descriptions, results on the realization problem, and solutions to the problem of regulation by feedback. Most of these results are in fact based on extensions of a certain realization of matrix description representations introduced by Fuhrmann [9], which follows the module theoretic realization theory of Kalman (see Kalman, Falb, and Arbib [13]).

Generalizations along this line to the discrete-time time-varying systems has been done by some authors (see e.g., Emre [5], Kamen and Hafez [15], Kamen, Khargonekar, and Poolla [17], and Khargonekar and Poolla [19], and the references there), where the module theoretic realization theory

---

[1] Part of the research of this author was supported in part by the ONR Research Contract N00014-86-K-0538, U.S. Air Force under AFOSR Contract 85-0186, and U.S. Army Grant DAAG-29-85-K-0099, through the Center for Mathematical System Theory, University of Florida, Gainesville, FL 32611.

has been used to explore several control problems and some system theoretic properties of the associated polynomial matrix factorizations. Continuous-time case has been considered in Emre [5]. It has been already shown in above references that in the discrete-time case transfer matrices can be defined as formal power series in the indeterminate z^{-1} (the left shift operator), with a very similar mechanism as that used in the time-invariant case. However, for continuous-time systems, so far, there does not appear to be a useful formal power series definition of transfer matrices resembling those in the time-invariant or discrete-time time-varying case. Thus such a theory had largely been missing for continuous-time case.

For the continuous-time time-varying systems, Ylinen [25] has obtained several results for systems of the form $A(p)y = B(p)u$ and their interconnections, in which A and B are defined over a skew polynomial ring with coefficients in a subring of C^∞ without zero divisors. More recently, Ilchmann, Nurnberger, and Schmale [11] and Ilchmann [10] consider skew polynomial matrices with coefficients in the field of meromorphic functions on an open interval and study the time-varying systems in differential operator representations. Using these, several results have been obtained. Nevertheless, power series representations of transfer matrices which has proved to be useful in many classes of systems, has not been explored by them. One of our main results here is to show how transfer matrices can be defined for continuous-time linear time-varying systems, and how such a useful power series representation plays a crucial role in extending the results available in the time-invariant and discrete-time time-varying cases.

It is shown here that a realization theory of linear continuous-time time-varying systems can be established starting from an abstract Kalman input-output map defined over a ring of skew polynomials consisting of time-varying differential polynomials. Transfer matrices can be defined as formal power series as in the time-invariant case. They are shown to be equivalent to impulse response in time-domain representations. Thus, we obtain a realization theory very analogous to the time-invariant case. This also clarifies the relations between the transfer matrix and the time-domain impulse response matrix and establishes fractional representations. Within this framework, an extended realization of fractional representations is also given here.

2. PRELIMINARIES

Let K be a non-trivial (arbitrary, fixed) subring of C^∞ real-valued functions closed under infinitely many derivatives defined on open subintervals of **R**, with the operations of pointwise addition and pointwise multiplication.

Let K[p] be the set of all time-varying differential polynomials

$$\sum_{i=0}^{n} p^i a_i$$

in the indeterminate p (which will also be interpreted as the derivative operator) with coefficients a_i in K. We define

$$pa = ap + (pa). \tag{2.1}$$

($(pa = \frac{d}{dt} a)$. Throughout this paper, $(p^k a) = \frac{d^k}{dt^k} a(t)$ denotes the k-th derivative of a(t) and p in $p^k a$ (without parenthesis) will stand for the indeterminate. K[p] is a noncommutative ring with identity, with coefficients written on the right. Thus K[p] is a skew polynomial ring. Here the term "skew" means that the coefficients of the polynomial do not necessarily commute with the indeterminate. For $b \in K$, $a(p) \in K[p]$, a(p)b is obtained by usual formal multiplication in $K[p]$, while $a(p)(b)$ represents the action of the differential operator $a(p)$ on b.

Next we introduce another indeterminate, p^{-1}. We constrain p and p^{-1} by

$$p^{-1}p = pp^{-1} = I, \tag{2.2}$$

where I is the identity operator. We also constrain p^{-1} via

$$p^{-1}a = ap^{-1} - p^{-1}(pa)p^{-1}, \text{ for } a \varepsilon K. \tag{2.3}$$

2.4. REMARK. The constraints (2.1) - (2.3) on the indeterminates p and p^{-1} are closely related. For example, it can be verified that (2.1) and (2.2) imply (2.3).

The skew ring of formal power series $K((p^{-1}))$ is defined by

$$\{ \sum_{i=-N}^{\infty} a_i^1 a_i^2 \dots a_i^{m_i}, \text{ where } a_i^j \varepsilon K \text{ or } a_i^j = p^{k_i},$$

$$\text{where, } \Sigma k_i, \text{ the total degree of } p \text{ equals } i \text{ for each } i \}.$$

Using (2.3) successively, we obtain

$$p^{-1}a = \sum_{i=1}^{\infty} (-1)^{i-1}(p^{i-1}a)p^{-i}; \ a \varepsilon K, \tag{2.4}$$

and

$$ap^{-1} = \sum_{i=1}^{\infty} p^{-i}(p^{i-1}a); \ a \varepsilon K. \tag{2.5}$$

Using these identities, (2.1) and (2.4-5), every element $W(t,p)$ of $K((p^{-1}))$ can be written uniquely in both forms below:

$$W(t,p) = \sum_{i=N}^{\infty} p^{-i}a_i; \ a_i \varepsilon K,$$

or

$$W(t,p) = \sum_{i=N}^{\infty} b_i p^{-i}; \ b_i \varepsilon K, \text{ where } N \text{ is some integer.}$$

The same is true for matrices over $K((p^{-1}))$, where now a_i, b_i are replaced by matrices over K. $W(t,p)$ is called (strictly) proper iff $N \geq 0$ ($N \geq 1$).

If x is an element of $K((p^{-1}))$, then $(x)_-$ denotes the strictly proper part of x and $(x)_{-1}$ represents the coefficient of p^{-1}. $p^{-1}K[[p^{-1}]]$ denotes the set of strictly proper formal power series. The matrices over a ring $K[p]$ of skew differential polynomials are called skew polynomial matrices and denoted as $A(t,p)$ with entries $a_{ij}(t,p)$. Each $A(t,p) \varepsilon K^{mxn}[p]$ can be uniquely written in the form

$$A(t,p) = A_0 + pA_1 + \dots + p^r A_r \quad \text{with } A_0, \dots, A_r \varepsilon K^{mxn}.$$

If $A_r \neq 0$, r is said to be the degree of $A(t,p)$ denoted as $\delta(A(t,p))$. The coefficient A_r of $A(t,p)$ of degree r is then called the leading coefficient matrix. In particular, $A(t,p)$ is monic if $m=n$ and $A_r = I_n$ = nxn identity matrix. $A(t,p)$ is called unimodular if there exists $B(t,p) \varepsilon K^{nxn}[p]$ such that $AB = I_n$ and $BA = I_n$. Two matrices $A(t,p)$, $B(t,p)$, $B(t,p) \varepsilon K^{mxn}[p]$ are said to be row equivalent if there exists a unimodular matrix $Q(t,p) \varepsilon K^{mxm}[p]$ such that $A(t,p) = Q(t,p)B(t,p)$. $A(t,p) \varepsilon K^{mxn}[p]$ and $B(t,p) \varepsilon K^{rxn}[p]$ are called right Bezout if there exist $S(t,p) \varepsilon K^{nxm}[p]$ and $T(t,p) \varepsilon K^{nxr}$ such that

$$S(t,p)A(t,p) + T(t,p)B(t,p) = I_n. \tag{2.6}$$

We now consider the left division in $K[p]$.

2.7. DEFINITION. A skew polynomial matrix $Q(t,p) \varepsilon K^{nxn}[p]$ is said to be admissible if there exists a unique formal power series

$$Q^{-1}(t,p) = \sum_{i=-M}^{\infty} (p^{-1})^i A_i = \sum_{i=-M}^{\infty} B_i (p^{-1})^i \quad \text{with } A_i, B_i \in K^{nxn}$$

such that

$$Q^{-1}(t,p)Q(t,p) = Q(t,p)Q^{-1}(t,p) = I.$$

Examples of admissible $Q(t,p)$ are ones with invertible leading coefficient matrix.

2.8. LEMMA. Let $Q(t,p)$ be an admissible skew polynomial matrix. Then to any nxm skew polynomial matrix $L(t,p)$, there corresponds a unique pair of skew polynomial matrices $B(t,p)$ and $A(t,p)$ such that

$$L(t,p) = Q(t,p)A(t,p) + B(t,p)$$

$$\text{and } Q^{-1}(t,p)B(t,p) \text{ is strictly proper.}$$

Let $K^n[p]$ be the set of all nx1 skew polynomial operators. For an nxn admissible matrix $Q(t,p)$ over $K[p]$, K_Q denotes $\pi_Q(K^n[p])$ where

$$\pi_Q : K^n[p] \to K_Q$$

$$: x(t,p) \to Q(t,p)(Q^{-1}(t,p)x(t,p))_- = x(t,p) \bmod Q(t,p).$$

$K^n[p]$ is clearly a right module over the noncommutative ring $K[p]$ where multiplication is operator composition. For a polynomial (t,p) in $K[p]$, it is easily checked that K_Q is a right $K[p]$-module if we define multiplication as

$$x(t,p)\alpha(t,p) := \pi_Q(x(t,p)\alpha(t,p)).$$

In addition, π_Q is a right $K[p]$-module homomorphism (i.e., linear map) whose kernel is $Q(t,p)K^n[p]$.

Note that all the above-presented can be dualized using row vectors and interchanging the words "left" and "right". We can define right division, etc., similarly (dual way), and thus a dual π_Q^d which is a left $K[p]$-module homomorphism.

3. REALIZATION (FRACTIONAL REPRESENTATIONS AND TRANSFER MATRICES)

In this section, we develop the theory of the polynomial fractional representation approach to linear continuous-time time-varying systems. The algebraic objects are noncommutative rings of polynomials, and formal power series, with coefficients in a subring of $C^\infty(0,\infty)$.

Similar to the time-invariant case established by Kalman (see Kalman, Falb, and Arbib [13]), we define the input space by

$$K^m[p] := \{u_0 + pu_1 + \ldots + p^k u_k : u_i \in K^m\},$$

i.e., a right $K[p]$-module of polynomials in the indeterminate p with coefficients in K^m. On the other hand, the output space is given as

$$p^{-1}K^r[[p^{-1}]] := \{ \sum_{i=1}^{\infty} (p^{-1})^i y_i : y_i \in K^r\}.$$

In other words, the output space is a skew module of strictly proper formal power series in the indeterminate p^{-1}.

The abstract right Kalman Input-Output map is defined in the following.

3.1. DEFINITION. Let m and r be positive integers. A linear right I-O map f is a right $K[p]$-module homomorphism

$$f : K^m[p] \to p^{-1}K^r[[p^{-1}]]$$

where

$$f[u(t,p)\alpha(t,p)] := ([f(u(t,p))]\alpha(t,p))_-,$$

where $u(t,p) \in K^m[p]$ and $\alpha(t,p) \in K[p]$ and $f(u(t,p))\alpha(t,p)$ represents the ordinary product of a formal power series by a polynomial.

Defining $W(t,p)$ to be $f(I)$, it is seen that for any input-output map - as defined above, there exists an rxm matrix function $W(t,p)$ such that

$$f(u(t,p)) = (W(t,p)u(t,p))_-$$

for any $u(t,p) \varepsilon K^m[p]$. The matrix function $W(t,p)$ is the causal formal transfer matrix associated with f and can be written as

$$W(t,p) = p^{-1}A_1 + (p^{-1})^2A_2 + \ldots.$$

where $A_i \varepsilon K^{rxm}$, $i=1,2,\ldots$, are the impulse response coefficient matrices. It can be seen that $W(t,p)$ is the rxm matrix over $p^{-1}K[[p^{01}]]$. Then the input-output equation $y(t,p) = f(u(t,p))$ can be characterized in terms of the impulse response coefficient matrices A_i as follows.

With $u(t,p) = u_0 + pu_1 + \ldots + p^ku_k$ as input,

$$f(u(t,p) = \sum_{i=0}^{k} [(E-pI)^i([A_1^T, A_2^T, \ldots]^T)]u_i = p^{-1}y_1 + p^{-2}y_2 + \ldots,$$

where

$$E = \begin{bmatrix} 0 & I_r & 0 & \ldots \\ 0 & 0 & I_r & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

In matrix form, $-(u(t,p))$ is represented in terms of ordinary matrix multiplication as

$$\left[\begin{bmatrix} A_1 \\ A_2 \\ \vdots \end{bmatrix} : (E-pI)(\begin{bmatrix} A_1 \\ A_2 \\ \vdots \end{bmatrix}) : \ldots : (E-pI)^k(\begin{bmatrix} A_1 \\ A_2 \\ \vdots \end{bmatrix})\right] \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}.$$

Define

$$\Gamma = [\tilde{A} : (E-pI)(\tilde{A}) : \ldots : (E-pI)^k(\tilde{A}) : \ldots :] \tag{3.3}$$

where

$$\tilde{A} = [A_1^T, A_2^T, \ldots]^T. \tag{3.4}$$

We call the infinite matrix Γ the Hankel matrix associated with the linear continuous-time time-varying input-output map f. Later, we will show that this is in fact the Hankel map first introduced by Kalman (Kamen [14], Kalman [12]). Γ is a matrix with entries in K.

We define a finite dimensional system $\Sigma = (F,G,H)$ to be a realization of the (right) I-O map f (or, equivalently, of $W(t,p)$ or equivalently, of Γ) iff

$$A_i = [(pI+F^T)^{i-1}(H^T)]^TG;\ i=1,2,\ldots.$$

It can be shown in a straightforward way that Γ is realizable iff

$$\Gamma = \begin{bmatrix} H \\ [(pI+F^T)(H^T)]^T \\ [(pI+F^T)^2(H^T)]^T \\ \vdots \end{bmatrix} [G, (F-pI)(G), \ldots, (F-pI)^k(G), \ldots].$$

We call the matrix $[H^T, (pI+F^T)(H^T), \ldots]^T$ the observability map associated with Σ denoted as O_Σ, and the matrix $[G, (F-pI)(G), \ldots, (F-pI)^k(G), \ldots]$ the reachability map associated with Σ, denoted by C_Σ. Then

$$\Gamma = O_\Sigma C_\Sigma.$$

Dually, *a left input-output map $\mathscr{f}_d$ (to distinguish it from a right I-O map as defined above)* can be defined as a left K[p]-module homomorphism as follows. Writing input space elements as row vectors

$$u(t,p) = u_0 + u_1 p + \dots + u_k p^k$$

(where $u_i \in K^r$ are row vectors) and the output space elements as row vectors

$$y(t,p) = y_1 p^{-1} + y_2 p^{-2} + \dots$$

(y_i is a row vector in K^m), we can define an I-O map as a left K[p]-module homomorphism between these two spaces.

Then we can define the *associated transfer matrix* and Hankel map Γ_d similarly.

3.5. PROPOSITION. Let $\mathscr{f}$ be a right I-O map with transfer matrix W(t,p) and the Hankel matrix Γ. Let $\mathscr{f}_d$ be the dual left I-O map with the transfer matrix $W_d(t,p)$ and Hankel map Γ_d. Then

i) $\Gamma_d = \Gamma$,

ii) $$W(t,p) = \sum_{i=1}^{\infty} p^{-i} A_i = W_d(t,p)$$

$$= \sum_{i=1}^{\infty} B_i p^{-i},$$

where $[A_1^T, A_2^T, \dots, \dots]^T$ is the first block column of Γ, and $[B_1, B_2, \dots]$ is the first block row of Γ.

We call a left I-O map *realizable* by a finite dimensional system iff there exists a finite triple (F,G,H) such that

$$B_i = H(F-pI)^{i-1}(G);\ i = 1,2,\dots.$$

These combined with results of Kamen [14] (using the technique in Kamen [14, Prop. 7] without any resort to impulse response yet, and without the freeness constraint imposed there) yield

3.6. THEOREM.

a) A right (left) I-O map $\mathscr{f}$ is realizable *by a finite dimensional system* iff the K-module generated by columns (rows) of its Hankel map is a submodule of a finitely generated K-module.

b) Every right I-O map defines a dual (left) I-O map via its Hankal map and vice versa (i.e., both have the same Hankel maps). $\mathscr{f}$ is realizable iff $\mathscr{f}_d$ is realizable; a realization of $\mathscr{f}$ is also a realization of $\mathscr{f}_d$, and vice versa.

c) (F,G,H) is a realization of $\mathscr{f}$ (or $\mathscr{f}_d$) iff $W(t,p) = H(pI-F)^{-1}G$.

p^{-1} has been so far considered as an indeterminate as defined in Section II. Now, if we interpret p^{-1} as $\int_0^t d\zeta$, then for any integrable function f, we have the following identity (see Kecs [18]).

$$(p^{-1})^n f = \int_0^t f(\zeta) \frac{(t-\zeta)^{n-1}}{(n-1)!} d = f(t) * \frac{t^{n-1}}{(n-1)!}. \tag{3.7}$$

Applying (3.7) to the input-output equation $y = \mathscr{f}(u)$ in terms of the transfer matrix W(t,p), we obtain

$$y(t) = W(t,p)u(t) = \int_0^t A(t,\zeta)u(\zeta)d\zeta$$

where $A(t,\zeta)$ is the causal impulse response matrix, and convergence conditions (such as analyticity) are assumed. Also, if for a fixed ζ, we expand $A(t,\zeta)$ into a Taylor series about $t=\zeta$, this yields

$$A(t,\zeta) = \sum_{i=1}^{\infty} \left(\frac{\partial^{i-1} A(t,\zeta)}{\partial t^{i-1}}\bigg|_{t=\zeta}\right) \frac{(t-\zeta)^{i-1}}{(i-1)!} = \sum_{i=1}^{\infty} A_i(\zeta) \frac{(t-\zeta)^{i-1}}{(i-1)!}.$$

Hence A_i's are the coefficients of the Taylor series expansion of the impulse response matrix about $t=\zeta$. It is seen that the transfer matrices introduced here are in direct correspondence with the impulse responses in time-domain.

Dually, with $W(t,p) = \sum_{i=1}^{\infty} B_i(p^{-1})^i$, (transfer matrix expressed in accordance to the dual I-O map)

$$y(t) = W(t,p)u(t) = \int_0^t A(t,\zeta)u(\zeta)d\zeta$$

via Taylor series representation, where

$$B_i(t) := \left.\frac{\partial^{i-1}}{\partial t^{i-1}} A(t,\zeta)\right|_{\zeta=t.}$$

Note that for C^{∞} systems Taylor series is always well defined and the above correspondence with transfer matrices can always be established.

This shows that

3.8. THEOREM. The Hankel map associated with an abstract I-O map is the same Hankel map as the one obtained from the impulse response. The realization $\Sigma = (F,G,H)$ of a realizable abstract I-O map as defined above, in fact, is the realization of the impulse response in the usual sense (see Kamem [14] for this latter situation). Transfer matrices over C^{∞} are in 1-1 correspondence with C^{∞} impulse responses which are in 1-1 correspondence with realizable Kalman I-O maps.

Next we note

3.8a. LEMMA. Let

$$Q(t,p) = Ip^k + D_{k-1}p^{k-1} + \ldots + D_1 p + D_0.$$

Then Q is admissible.

3.9. DEFINITION. Let $W(t,p)$ be the transfer matrix associated with a right (or left) I-O map. Let Γ be the associated Hankel map. We call $W(t,p)$ right (left) rational iff $W(t,p) = P_1(t,p)Q_1^{-1}(t,p)$ $(= Q^{-1}(t,p)R(t,p))$ for polynomials P_1, Q_1, Q, R with Q_1, Q monic.

3.10. THEOREM. i) A transfer matrix $W(t,p)$ is left (right) rational iff the rows (columns) of Γ form a finitely generated K-module.

ii) $W(t,p)$ is left (right) rational iff $W(t,p) = Q^{-1}R$ $(W(t,p) = P_1 Q_1^{-1})$ for some polynomials Q, R, P_1, Q_1, with Q and Q_1 admissible.

We define the reachability index (observability index) $\mu(v)$ of a system (F,G,H) over K as the first integer k such that

$$(F-pI)^k(G) \subset \text{im } G + \ldots + \text{im } (F-pI)^{k-1}(G)$$

$$((F^T+pI)^k(H^T) \subset \text{im } H^T + \ldots + \text{im } (F^T+pI)^{k-1}(H^T))$$

provided such a finite integer exists.

3.11. COROLLARY. A system (F,G,H) over K has a finite reachability (observability) index iff its transfer matrix $W(t,p)$ is right (left) rational; iff $W(t,p)(\Gamma)$ has a reachable realization.

With the results on formal power series established here, the next result can be proved the same way as in Khargonekar and Poolla [20, Proposition 2.4].

3.12. THEOREM. A polynomial matrix $Q(t,p)$ is admissible iff there exist polynomial matrices $M(t,p)$, $M_1(t,p)$ such that $M(t,p)Q(t,p)$ and $Q(t,p)M_1(t,p)$ are both monic.

Thus, it is seen that if a polynomial matrix $Q(t,p)$ is admissible, then it is both left and right admissible in the sense of Emre [5].

Next we extend the realization of Fuhrmann [9] to the realization of the strictly proper part of a transfer matrix

$$W(t,p) = P(t,p)Q^{-1}(t,p)R(r,p)$$

with Q admissible.

Define

$$R_1(t,p) := \pi_Q(R(r,p))$$

and

$$\hat{G} : K^m \to K_Q; \qquad \hat{G}u := R_1(t,p)u,$$

$$\hat{F} : K_Q \to K_Q; \qquad \hat{F}(x) := \pi_Q(xp),$$

$$\hat{H} : K_Q \to K^r; \qquad \hat{H}(x) := (P(t,p)Q^{-1}(t,p)x)_{-1},$$

where K_Q and π_Q are defined as in Section II. Note that K_Q is both a right K-module, and a right K[p] module. π_Q is both a right K and K[p] module homomorphism. With these definitions, one can verify that for any $u \in K^m$, and integer $k \geq 0$,

$$\pi_Q(R(t,p)p^k u_k) = \hat{F}^k(\hat{G})\cdot u_k.$$

Thus for any $u(t,p) = \sum_{i=0}^{k} p^i u_i \in k^m[p]$,

$$\pi_Q(R(t,p)u(t,p)) = \sum_{i=0}^{k} [\hat{F}^i(\hat{G})]u_i. \tag{3.13}$$

3.14. REMARK. One can dualize the results already established. If $Q(t,p)$ is an admissible skew polynomial matrix, then for any $u(t,p)$, $(Q^{-1}(t,p)u(t,p))_-$ defines a right I-O map, which is purely defined by $(Q^{-1}(t,p))_-$. Thus, via its Hankel map, $(Q^{-1}(t,p))_-$ can be expressed as $\sum_{i=1}^{n} \tilde{Q}_i(p^{-1})^i$ for some $\tilde{Q}_i$ over K. Thus, we can now define the associated (dual) left I-O map and define

$$\pi_Q^d(u(t,p)) := [(u(t,p)Q^{-1}(t,p))_-]Q(t,p).$$

Hence we can express any skew polynomial matrix $M(t,p)$ uniquely through the right division as

$$M(t,p) = C(t,p)Q(t,p) + D(t,p); \quad DQ^{-1} \text{ strictly proper.}$$

Then one can dualize the results in terms of left module structure, left module homomorphisms.

Thus, we see that given a transfer matrix $W(t,p)$ expressed in a fractional form

$$W(t,p) = PQ^{-1}R, \tag{3.15}$$

with Q admissible, the associated I-O map.

$$f(u(t,p)) = (W(t,p)u(t,p))_-$$

can be factorized as

$$f = h\circ g \tag{3.16}$$

where

$$g = \pi_Q : K^m[p] \to K_Q$$

and

$$h : K_Q \to p^{-1}K^r[[p^{-1}]]; \; x(t,p) \to (PQ^{-1}x(t,p]))_-.$$

It can be checked that if

$$(W(t,p))_- = \sum_{i=1}^{\infty} B_i p^{-i},$$

then

$$B_i = \hat{H}(\hat{F}^{i-1}(\hat{G})); \; i = 1, 2, \ldots.$$

3.17. DEFINITION. Let $W(t,p)$, f, g, h be as in (3.15) and (3.16). Then we call the Q-realization $= (\hat{F}, \hat{G}, \hat{H})$ of $(W(t,p))_-$ reachable iff π_Q is onto. Σ is observable iff h is 1-1.

The following provides an extension of the techniques in Emre [1,5] to solve Bezout type equations.

3.18. THEOREM. Let Q, R, Φ be polynomial operators with Q admissible. There exist polynomial time-varying operators $X(t,p)$, $Y(t,p)$ such that

$$Y(t,p) = p^k Y_k + \ldots + pY_1 + Y_0,$$

and

$$QX + RY = \Phi$$

iff Y is given as a solution of linear equations over K

$$[\hat{G}, \hat{F}(\hat{G}), \ldots, \hat{F}^k(\hat{G})][Y_0^T, Y_1^T, \ldots, Y_K^T]^T = \pi_Q(\Phi)$$

where $(\hat{F}, \hat{G}, \hat{H})$ is the Q-realization of $(Q^{-1}R)_-$.

3.19. REMARK. This theorem can be easily dualized to obtain solution of $XQ + YP = \Phi$ in terms of the dual Q-realization of $PQ^{-1}(\pi_Q^d, \hat{F}^T, \hat{H}^T, \hat{G}^T)$ and reachability map π_Q^d defined in Remark 3.14.

It is easy to infer the following corollary.

3.20. COROLLARY. With conditions of Theorem 3.18, there exist X, Y such that $QX + RY = I$ iff the Q-realization of $(Q^{-1}R)_-$ is reachable.

3.21. THEOREM. K_Q (K_Q^d) is a finitely generated right (left) K-module.

3.23. COROLLARY. Suppose in Theorem 3.18, in addition, $Q^{-1}R$ is also right rational. Then the integer k occurring in (3.18a) can be always chosen to be $\mu-1$, where μ is the reachability index of $(\hat{F}, \hat{G})$ in (3.18a) ((F,G) in (3.22)).

3.24. THEOREM. Let Q be admissible, and let $W = Q^{-1}R$ be given. Suppose Q, R satisfy the Bezout identity (i.e., $QX + RY = I$ for some polynomials X, Y), (or equivalently the Q-realization of W is reachable). Under these conditions, W is right rational.

REFERENCES

[1] E. Emre, "The polynomial equation $QQ_c + RP = \Phi$ with application to dynamic feedback," SIAM J. Contr. Optim. 18, 611-620, 1980.

[2] E. Emre, "On necessary and sufficient conditions for regulation of linear systems over rings," SIAM J. Contr. Optim. 18, 611-620, 1982.

[3] E. Emre, "Regulation of linear systems over rings by dynamic output feedback," Systems & Control Lett. 3, 57-62, 1983.

[4] E. Emre, "Generalized model matching and (F,G)-invariant submodules for linear systems over rings," Linear Algebra & its Appl. 50, 133-166, 1983.

[5] E. Emre, "A polynomial fractional approach to linear time-varying systems," Proc. 21st Allerton Conf., 11-20, 1983.

[6] E. Emre, "Observers for nonlinear/time-varying systems," Proc. 21st Allerton Conf., 564-572, 1983.

[7] E. Emre and P.P. Khargonekar, "Regulation of split linear systems over rings: coefficient-assignment and observers," IEEE Trans. Autom. Contr. AC-27, 104-113, 1982.

[8] E. Emre, H.-M. Tai and J.H. Seo, "Fractional representation, transfer matrices and control of continuous-time linear time-varying systems," Preprint, EE Dept., Texas Tech Univ., 1987, Submitted for publication.

[9] P.A. Fuhrmann, "Algebraic system theory: an analyst's point of view," J. Franklin Institute 301, 521-540, 1976.

[10] A. Ilchmann, "Time-varying linear systems and invariants of system equivalence," Int. J. Contr. 42, 759-362, 1984.

[11] A. Ilchmann, I. Nurnberger and W. Schmale, "Time-varying polynomial matrix systems," Int. J. Contr. 42, 329-362, 1984.

[12] R.E. Kalman, private communication, 1986.

[13] R.E. Kalman, P.L. Falb and M.A. Arbib, Topics in Mathematical System Theory, McGraw-Hill, New York, 1969.

[14] E.W. Kamen, "New results in realization theory for linear time-varying analytic systems," IEEE Trans. Autom. Contr. AC-24, 866-877, 1979.

[15] E.W. Kamen and K.M. Hafez, "Algebraic theory of linear time-varying systems," SIAM J. Contr. Optim. 17, 500-510, 1979.

[16] E.W. Kamen and P.P. Khargonekar, "A transfer function approach to linear time-varying discrete-time systems," Proc. 21st CDC, 152-157, 1982.

[17] E.W. Kamen, P.P. Khargonekar and K.R. Poolla, "A transfer function approach to linear time-varying discrete-time systems, SIAM J. Contr. Optim. 23, 550-565, 1985.

[18] W. Kecs, The Convolution Product and Some Applications, Kluwer, Boston, 1982.

[19] P.P. Khargonekar, "On matrix fraction representations for linear systems over commutative rings," SIAM J. Contr. Optim. 20, 172-197, 1982.

[20] P.P. Khargonekar and K.R. Poolla, "Polynomial matrix-fraction representations for linear time-varying systems," Linear Algebra & its Appl. 80, 1-37, 1986.

[21] H.H. Rosenbrock, State-Space and Multivariable Theory, John Wiley & Sons, New York, 1970.

[22] L.M. Silverman, "Realization of linear dynamical systems," IEEE Trans. Autom. Contr. AC-16, 554-567, 1971.

[23] L. Weiss and R.E. Kalman, "Contributions to liner system theory," Int. J. Eng. Sci. 3, 141-171, 1965.

[24] W.M. Wonham, Linear Multivariable Control: A Geometric Approach, Springer-Verlag, Berlin, 1979.

[25] R. Ylinen, "An algebraic theory for analysis and synthesis of time-varying linear differential systems," Acta Polytechnica Scandinavica Ma. 32, 2-61, 1980.

CONDITIONS FOR EXISTENCE OF AN OPTIMAL CONTROL IN THE LQ REGULATOR PROBLEM FOR SINGULAR SYSTEMS

J. DANIEL COBB
Department of Electrical and Computer Engineering
University of Wisconsin
1415 Johnson Drive
Madison, WI 53706-1691

INTRODUCTION

We consider the problem of minimizing the cost functional

$$J(x,u) = \int_0^\infty [x^T \; u^T] \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt \tag{1}$$

subject to the singular system constraint

$$E\dot{x} = Ax + Bu \tag{2}$$

where $E,A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, E is singular, $R > 0$, and $\begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \geq 0$. Certain fundamental questions need to be answered concerning the relationship between the matrices Q, N, R, P, E, A, and B and the optimal pairs (x^*,u^*). The most obvious of these follow.

1) Under what conditions does there exist at least one optimal (x^*,u^*) and when is it unique?
2) To what function spaces do the (x^*,u^*) belong?
3) When can an optimal control u^* be implemented according to a feedback law on x^*?
4) How can one compute the optimal pairs efficiently?

For the case $N = 0$ and $Q > 0$, we answered questions 1)-3) in [1]. Question 4) is treated for the general case by Bender and Laub in [2] using necessary conditions derived by Jonchkeere [3]. Jonchkeere also investigates conditions under which his necessary conditions exhibit existence and uniqueness of solutions. The question of existence of an optimal pair (x^*,u^*) when $N \neq 0$ or when Q is singular has remained open and motivates our present work.

As a first attempt at a solution, we reformulate the optimization problem by "completing the square" in $J(x,u)$. Let $\hat{u} = u + R^{-1}N^Tx$, $\hat{Q} = Q - NR^{-1}N^Tx$, and $\hat{A} = A - BR^{-1}N^T$. Then we may rewrite the cost (1) and system constraint (2) as

$$J(x,\hat{u}) = \int_0^\infty x^T\hat{Q}x + \hat{u}^TR\hat{u}\, dt, \quad E\dot{x} = \hat{A}x + B\hat{u} \tag{3}$$

This yields an equivalent form which can be handled by the methods of [1] as long as $\hat{Q} > 0$ and $\det(sE - \hat{A}) \not\equiv 0$. If either condition fails to hold, some new theory is required.

HILBERT SPACE FORMULATION

We first note that, as in [1], we must take $J = \infty$ whenever the integrand of (3) is not in L^2 (e.g. if it contains impulses). Let $\mathscr{D}_+$ be the set of all distributions with support in $[0,\infty)$ and let $\mathscr{D}_0$ be the subspace of $\mathscr{D}_+^{n+m}$ consisting of all elements z satisfying $Cz = 0$, where $C^TC = \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix}$. Also let $\bar{\mathscr{D}}$ be the quotient space of $\mathscr{D}_+^{n+m}$ with respect to $\mathscr{D}_0$,

$$<[z_1],[z_2]> = \int_0^\infty z_1^T C^T C z_2 \, dt \qquad (4)$$

where $[z] \in \bar{\mathscr{D}}$ is the equivalence class determined by z, and $\mathscr{L}_2[0,\infty) = \{d \in \bar{\mathscr{D}} \mid \bar{J}(q) < \infty\}$. We can prove the following

Lemma Equipped with the inner product (4), $\mathscr{L}_2[0,\infty)$ is a Hilbert space.

Consider now the system equation (2). We define $\mathscr{N}(x_0) \in \bar{\mathscr{D}}$ to be the equivalence class generated by the natural response of (2) due to the initial condition x_0. Also let

$$\Lambda = \{[z] \mid z = \begin{bmatrix} x \\ u \end{bmatrix} \text{ and x is the forced response of (2) due to u}\}$$

The original optimization problem is easily seen to be equivalent to the problem of minimizing $\bar{J}(w) = <w,w>$ over $\mathscr{N}(x_0) + \Lambda$.

Theorem If the system (2) is stabilizable and impulse controllable (see [ ]), $(\mathscr{N}(x_0) + \Lambda) \cap \mathscr{L}_2[0,\infty) \neq \phi$ and $(\mathscr{N}(x_0) + \Lambda) \cap \mathscr{L}_2[0,\infty)$ is a closed linear variety in $\mathscr{L}_2[0,\infty)$.

The Hilbert space projection theorem thus implies that a unique optimum exists in $\mathscr{L}_2[0,\infty)$ under the same conditions as stated in [1]. Hence, an optimal pair (x^*,u^*) exists for the original problem under these same conditions. The pair (x^*,u^*) in general is not unique, however, since the optimum $w^* \in \mathscr{L}_2[0,\infty)$ may be generated by many solutions pairs of (2).

[1] J. D. Cobb, "Descriptor Variable Systems and Optimal State Regulation", *IEEE Transactions on Automatic Control*, Vol. 28, No. 5, May 1983, 601-611.

[2] D. J. Bender, A. J. Laub, "The Linear-Quadratic Optimal Regulator for Descriptor Systems," *IEEE Transactions on Automatic Control*, in press.

[3] E. Jonckheere, "Variational Calculus for Descriptor Systems," *IEEE Transactions on Automatic Control*, in press.

STABILITY PARAMETERIZATIONS AND HIDDEN MODES IN TWO DEGREES OF FREEDOM CONTROL DESIGN

P. J. Antsaklis and O. R. González
Dept. of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, IN 46556

ABSTRACT

The general two degrees of freedom controller provides a unifying framework in approaching complicated control problems involving multiple objectives, in a manner independent of specific, and possibly restrictive, control configurations. A stability theorem is used which clearly shows the role of the feedback part of the controller and it allows the introduction of alternative stabilizing controller parameterizations in a natural way. The stable uncontrollable and/or unobservable hidden modes of the control system are then characterized and expressed in terms of the design parameters; this leads to strategies to avoid the hidden modes if so desired and in general it leads to better, more efficient control design.

INTRODUCTION

The objective of any multivariable control design method is to synthesize an appropriate control law, represented here by the multivariable controller C. It was noted earlier [1,2] that, for single-input, single-output plants, a single degree of freedom control configuration, such as error (unity) feedback, imposes restrictions on the attainable control properties. Not surprisingly, similar restrictions are present in the multi-input, multi-output case (e.g. [3]). This is of course to be expected, since an error feedback controller generates the plant input u by processing only the error r-y, where r is the reference (set-point, command) external input and y the plant output (the sensor output); that is, the two natural degrees of freedom, corresponding to the availability of the signals r and y, are not fully utilized by a single degree of freedom control configuration [4]. These two signals should be processed independently if unnecessary constraints on the attainable control objectives are to be avoided.

Our interest here is in linear plants P and linear controllers C. The general linear controller can be written as $u = -C_y y + C_r r$ where $[-C_y, C_r] = C$, with C_y, C_r transfer matrices to be determined. This controller, called two degrees of freedom controller, has been utilized by several researchers [4-11] and recently, parametric characterizations of all internally stabilizing controllers C have been derived and used in the design of control systems [4,7,8,12-14]. The two degrees of freedom controller provides a unifying framework in approaching complicated control problems involving multiple objectives, in a manner which is independent of specific, and possibly restrictive, control configurations. There is much renewed interest in the two degrees of freedom controller mainly due to more demanding control problems but also due to recent advances in understanding and effectively utilizing such control laws.

In this paper, a stability theorem is introduced first. It separates the effect of C_y, the feedback portion of the controller C, and it clearly shows that stability in two degrees of freedom control design is based on the stability of the well studied single degree of freedom configuration and it is an extension of previous results (due to $C_r \neq C_y$) as expected. Expressing the stability conditions in terms of (i) the single degree of freedom (C_y) stability condition, and (ii) a condition on C_r given C_y to guarantee internal stability, naturally allows the use of any single degree of freedom stabilizing feedback parameterization used in the literature. In other words, we build upon the single degree of freedom stability results to generate the two degrees of freedom results. Furthermore, this theorem directly leads to parametric characterizations of all input-output maps attainable with internal stability from r, the vector of request inputs; also to controller configurations that attain these maps.

The hidden modes of a compensated system correspond to the compensated system's eigenvalues which are uncontrollable and/or unobservable. The hidden modes for single degree of freedom controllers have been studied in the literature [3,15-17]. On the other hand, the hidden modes for two degrees of freedom controllers have not been studied in the literature except for in special cases [15,16,18]. In this paper we consider the hidden modes of the two degrees of freedom control system in Figure 1. It is assumed that the plant and controller are controllable and observable. In this way the hidden modes of the compensated system are due exclusively to the system interconnection. Later this assumption is relaxed to consider the implications of non-irreducible realizations of the controller.

Observe that the controller C is designed to guarantee at least internal stability of the system. Therefore the hidden modes of the compensated system, if any, will be stable. Stable hidden modes need to be studied because they can increase the necessary order of the controller and they can degrade the performance. Consequently, a complete understanding of stable hidden modes will lead to better control design algorithms.

STABILITY THEOREM. PARAMETERIZATIONS

The two degrees of freedom linear controller

$$u = C \begin{bmatrix} y \\ r \end{bmatrix} = [-C_y \quad C_r] \begin{bmatrix} y \\ r \end{bmatrix}, \tag{1}$$

where $C = [-C_y, C_r]$ proper, transfer matrices, generates the plant input u by independently processing the plant output y and the external reference input r (Figure 1),

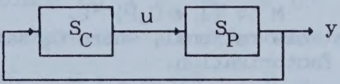


Figure 1. The compensated system.

where S_P is the linear plant described by y = Pu with P its proper transfer matrix and S_C is the controller described in (1). Several researchers have utilized C in a time domain state-space formulation (e.g. Bengtsson [5]). Using a transform domain formulation, C has been incorporated by Pernebo [6] and more recently, by Youla and Bongiorno [4], Desoer and Gustafson [7], Vidyasagar [8], Desoer and Gündes [10] and Sugie and Yoshikawa[11]. The pole-placement algorithm of Åström in [9] also uses C but for SISO plants only.

A significant step towards better understanding the role of C in plant compensation was recently accomplished by parametrically characterizing all stabilizing two degrees of freedom controllers C; thus extending the results on parametric characterization of all feedback controllers C_y [19-23,3,15,12-14] which have greatly contributed to control design methods

[19-20,24-28]. All internally stabilizing controllers C were parametrically characterized in [4,7-8] using two independent proper and stable parameters K and X as

$$C = (x_1 - K\tilde{N})^{-1}[-(x_2 + K\tilde{D}), \ X], \tag{2}$$

where $\tilde{N}$, $\tilde{D}$, x_1, x_2 are proper and stable transfer matrices, and they are derived from coprime fractional representations of the plant

$$P = ND^{-1} = \tilde{D}^{-1}\tilde{N}, \tag{3}$$

and the associated Bezout-Diophantine equation

$$x_1 D + x_2 N = I. \tag{4}$$

In [4], (2) involves polynomial matrices $\tilde{N}$, $\tilde{D}$, x_1, and x_2 with K and X stable transfer matrices; x_1 and x_2 satisfy the Diophantine equation $x_1D+x_2N = D_0$, where D, N and D_0 are polynomial matrices. Note that the development in [4] follows [19], while in [7-8] the development is based on the seminal paper of Desoer, et. al. [22], which among other results, it extends the results of [19,21] to more general (beyond polynomial) rings. The parameter K in (2) is the well known parameter used in the characterization of all stabilizing feedback controllers C_y in [19,22]. Note that the parameter X is actually the response parameter used by Antsaklis and Sain [3] (and Liu and Sung [29]) to parametrically characterize feedback controllers in an error feedback setting. If Dz = u, y = Nz is an internal polynomial matrix representation of the plant P, then it can be shown that

$$z = Xr, \tag{5}$$

that is, X is the transfer matrix between the external input r and the partial state z of the plant.

It is evident that if exogenous signals (such as disturbances, noise) are assumed to be injected at various points in Figure 1, all possible transfer functions and responses to all inputs can be derived in terms of K and X by direct substitutions of (2); in this way all "admissible", under internal stability responses can be characterized [4,7-8,14].

It is advantageous to study internal stability of the system in Figure 1 in a novel alternative way [30].

Theorem 1. Consider a proper controller C as in (1) and the system in Figure 1, and assume that $|I + C_yP| \neq 0$. The compensated system is internally stable if and only if

(i) $u = -C_y y$ internally stabilizes the system $y = Pu$, and

(ii) C_r is such that

$$M := (I + C_yP)^{-1}C_r \tag{6}$$

satisfies $D^{-1}M = X$, a stable rational, where C_y satisfies (i) and $P = ND^{-1}$ is a coprime polynomial factorization.

Theorem 1 provides some advantages over other stability theorems presented so far for the system in Figure 1. These advantages are discussed below and are followed by a proof of Theorem 1.

This theorem separates the role of C_y, the feedback part of C, from C_r in achieving internal stability. Clearly if only feedback action is considered, only (i) is of interest; and if open loop control is desired, $C_y = 0$, (i) implies that P must be stable, and $C_r = M$ must satisfy (ii). In (ii) the parameter M (=DX) appears rather naturally and in (i) the way is open to use any desired feedback parameterization, not necessarily K of [4,7-8].

From Theorem 1 we can directly characterize the input-output maps attainable from r with internal stability. In particular, consider the two maps described by y = Tr and u = Mr. The characterization is done in Theorem 2.

Theorem 2. A pair (T,M) is realizable with internal stability via a two degrees of freedom configuration if and only if (T,M) = (NX,DX) with X stable.

Proof of Theorem 2. Necessity. Assume (T,M) are realizable with internal stability. Consider $u = Mr$, (5), and (7). Then $Dz = Mr$ or $z = Xr$. Internal stability implies that X is stable.
Sufficiency. If X is stable, then T and M are stable. The only thing left to show is that a controller configuration exists to implement these maps. Consider the two degrees of freedom controller in Figure 2 ($C_r = \hat{M} + C_y\hat{T}$).

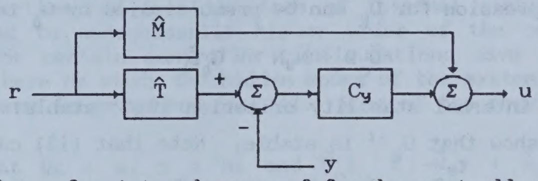


Figure 2. A two degrees of freedom controller.

This configuration implements the desired maps when $\hat{M} = M$ and $\hat{T} = T$. Other configurations to attain these maps are, of course, possible [14].

Q.E.D.

Proof of Theorem 1. Consider an irreducible polynomial internal description of the plant

$$Dz = u, \quad y = Nz, \tag{7}$$

and of the controller

$$\tilde{D}_c z_c = [-\tilde{N}_y \quad \tilde{N}_r] \begin{bmatrix} y \\ r \end{bmatrix}, \quad u = z_c, \tag{8}$$

where (N,D) and ($\tilde{D}_c$, $[-\tilde{N}_y, \tilde{N}_r]$) are coprime polynomial factorizations. With these definitions an internal description of the compensated system is

$$D_0 z = (\tilde{D}_c D + \tilde{N}_y N) z = \tilde{N}_r r, \quad y = Nz. \tag{9}$$

We say that if $|I + C_y P| \neq 0$ then the compensated sytem is internally stable if D_0^{-1} is stable.
Necessity: Assume that the compensated system is internally stable, that is, D_0^{-1} is stable. Since $C_y = \tilde{D}_c^{-1}\tilde{N}_y$ is not necessarily a coprime polynomial factorization, there exists a polynomial matrix G_ℓ so that

$$[\tilde{D}_c \quad \tilde{N}_y] = G_\ell [\tilde{D}_{C_y} \quad \tilde{N}_{C_y}], \tag{10}$$

where G_ℓ is a greatest common left divisor (g.c.l.d.) of $(\tilde{D}_c, \tilde{N}_y)$. Comparing (9) and (10) we notice that

$$\tilde{D}_{C_y} D + \tilde{N}_{C_y} N = G_\ell^{-1} D_0 := \tilde{D}_k, \text{ with } \tilde{D}_k^{-1} \text{ stable,} \tag{11}$$

where $\tilde{D}_k$ is a polynomial matrix. Observe that (11) also implies that G_ℓ^{-1} is stable. Hence (11) shows that $u = -C_y y$ stabilizes $y = Pu$, that is, part (i) of Theorem 1 is satisfied. To show (ii) write

$$\begin{aligned} M = (I + C_y P)^{-1} C_r &= D\tilde{D}_k^{-1}\tilde{D}_{C_y}(\tilde{D}_c^{-1}\tilde{N}_r) \\ &= D\tilde{D}_k^{-1} G_\ell^{-1}\tilde{N}_r \\ &= DX, \end{aligned} \tag{12}$$

where X := $D_o^{-1}\tilde{N}_r$ is a stable tranfer matrix. This shows that (ii) is necessary too.

Sufficiency: Let C satisfy (i) and (ii) of Theorem 1. If C = $\tilde{D}_c^{-1}[-\tilde{N}_y, N_r]$ is a coprime polynomial factorization and G_ℓ is a g.c.l.d. of $(\tilde{D}_c, \tilde{N}_y)$, then (10) is true for some left coprime matrices $\tilde{D}_{C_y}$ and $\tilde{N}_{C_y}$ ($C_y = \tilde{D}_{C_y}^{-1}\tilde{N}_{C_y}$). Because (i) is satisfied, $\tilde{D}_{C_y} D + \tilde{N}_{C_y} N = \tilde{D}_k$ is such that $\tilde{D}_k^{-1}$ is stable. The expression for $\tilde{D}_k$ can be premultiplied by G_ℓ to obtain

$$\tilde{D}_c D + \tilde{N}_y N = G_\ell \tilde{D}_k . \qquad (13)$$

In view of the internal stability criterion (D_o^{-1} stable) and $\tilde{D}_k^{-1}$ stable it suffices to show that G_ℓ^{-1} is stable. Note that (13) can be written as $G_\ell^{-1} D_o = \tilde{D}_k$ or as $G_\ell^{-1}\tilde{N}_r = \tilde{D}_k X$ with $(G_\ell, \tilde{N}_r)$ left coprime (if they were not coprime, C = $\tilde{D}_c^{-1}[-\tilde{N}_y, \tilde{N}_r]$ would not have been a coprime factorization). Therefore, $D_o = G_\ell \tilde{D}_k$ in (13) satisfies D_o^{-1} stable.

Q.E.D.

There are many choices in parametrically characterizing all feedback stabilizing controllers C_y and these are extensively discussed by Antsaklis and Sain in [15]. Parameterizations of all feedback stabilizing multivariable controllers were first introduced by Youla, Jabr and Bongiorno in [19]; also by Kucera [23]. Antsaklis [21] expressed the parameterization of Youla, et. al., over the polynomial ring, while Desoer, Liu, Murray and Saeks [22] extended the Youla parameterization over more general rings. Zames in [20] (see also [31,32]) introduced the parameter Q for the case of stable plants, and Antsaklis and Sain [3,33] discussed the parameter X ($=D^{-1}Q$, when an error feedback configuration is used) and derived conditions on X for the unstable plant case. Alternative parameterizations are discussed in [15]. The stabilizing controllers C can therefore be expressed, in addition to (2) as (for example):

$$C = (I - QP)^{-1}\,[-Q \quad M] = ((I - LN)D^{-1})^{-1}\,[-L \quad X], \qquad (14)$$

where Q = DL, M = DX with L, X stable and $D^{-1}(I - QP) = (I - LN)D^{-1}$ stable ($|I - QP| \neq 0$ or $|I - LN| \neq 0$), and $P = ND^{-1}$ a coprime polynomial matrix factorization. (14) gives parametric characterizations of all stabilizing controllers C, proper and nonproper. For C proper, M and Q are chosen proper and such that $(I - QP)^{-1}$ is also proper; note that if P is strictly proper, Q proper always implies that $(I - QP)^{-1}$ is proper. Notice that L or Q in (14) must satisfy certain conditions, in addition to being stable, in contrast to K in (2); however, alternative to K parameterizations, such as in (14), are very useful, since they do have certain additional desirable properties (see [15]).

HIDDEN MODES

The interconnection of a controller C, implemented via a minimal Kalman realization, controlling a minimal plant P as in Figure 1, might cause unobservable, uncontrollable modes from y, r respectively. In addition, when the controller C is put together by interconnecting separately designed subcontrollers, the overall controller's internal description might not be controllable and observable, even though the interconnected subcontrollers are assumed to be realized via minimal Kalman realizations. The uncontrollable and/or unobservable hidden modes, if not appropriately accounted for during the design, might lead to deterioration of system properties and even the loss of internal stability [15]; they could also lead to unnecessarily higher order of the controller. The hidden modes for certain controller configurations have been studied in [3,15,16,18]. Here we study the hidden modes of the system in Figure 1. A complete treatment is presented in [14]. Some of the results are outlined below:

Recall that $Dz = u$, $y = Nz$ and $\tilde{D}_c z_c = -\tilde{N}_y y + \tilde{N}_r r$, $u = z_c$ are irreducible descriptions of the plant and controller, respectively, assuming (N,D) and $(\tilde{D}_c, [-\tilde{N}_y, \tilde{N}_r])$ are coprime polynomial factorizations. Combining these two descriptions gives the following internal description of the feedback system in Figure 1

$$D_0 z = \tilde{N}_r r$$
$$\begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} N \\ D \end{bmatrix} z, \tag{15}$$

where $D_0 = \tilde{D}_c D + \tilde{N}_y N$. The roots of the determinant of D_0 are the closed-loop eigenvalues. The not necessarily irreducible internal description in (15) leads directly to the following preliminary characterization of hidden modes of the system in Figure 1: *The uncontrollable modes from r correspond to the poles of D_0^{-1} that cancel in $D_0^{-1}\tilde{N}_r$. The unobservable modes from y (u) correspond to the poles of D_0^{-1} that cancel in ND_0^{-1} (DD_0^{-1}).* The concept of cancellations in a product of transfer matrices has been clarified in [34-36,14]; the cancellations should be taken as pole cancellations rather than pole-zero cancellations.

A more useful characterization of hidden modes can be given in terms of the transfer matrices. The characterization is done in this way as in SISO systems because it gives insight into the mechanism of introducing hidden modes. These characterizations are also proven using internal descriptions. *The unobservable modes from y correspond to those poles of C which cancel in the product PC, and the unobservable modes from u correspond to those poles of P that cancel in $\tilde{N}_y P$. The uncontrollable modes from r correspond to the poles of $(I+C_yP)^{-1}$ that cancel in $(I+C_yP)^{-1}C_r$ and those poles of P that cancel in PM and C_yP.*. For comparison note that in the error feedback configuration $C_y = C_r$. In this case, the unobservable modes from y are the poles of C_y which cancel in PC_y whereas the uncontrollable modes from r are the poles of P which cancel in PC_y [3,15].

These results can be more useful in the design of the controller if they are written in terms of some of the design parameters, like L and X; in this way insight will be gained and the designer will be able to deal with the hidden modes using a variety of design tools and methodologies. Note that $L = D^{-1}(I + C_yP)^{-1}C_y$ and $X = D^{-1}(I + C_yP)^{-1}C_r$. *The unobservable modes from y correspond to the poles of [X, L] that cancel in N[X, L] and* *the unobservable modes from u correspond to the poles of* [X, L] *that cancel in* D[X, L] and to the poles of P that cancel in $D_0^{-1}[\tilde{N}_y, \tilde{N}_r]$. *The uncontrollable modes from r correspond to the poles of* P *that do not cancel in* $(I-LN)D^{-1}$ *and to the poles of* L *which are not poles of* X.

When a controller consists of the interconnection of separately designed subcontrollers, typically additional hidden modes are introduced because the resulting controller is not controllable and observable. The subcontrollers are designed to handle a particular problem such as stability and regulation. The hidden modes introduced depend on the particular interconnection. For example, when an observer is implemented with feedforward and feedback controllers, the poles of the observer appear as stable poles of the feedback controller and as stable zeros of the feedforward controller, and correspond to unobservable modes from y. It can be shown that the uncontrollable modes from $[y^t, r^t]^t$ (^{t} denotes transpose) of the controller become uncontrollable modes from r of the compensated system. Similarly, the unobservable modes from u of the controller become unobservable modes from y of the compensated system. It is also possible for the hidden modes of the controller to lead to other kinds of hidden modes of the compensated system. The characterization of these additional hidden modes also reduces to the well known case when $C_y=C_r$.

The above discussion on hidden modes is necessarily short. The results are based on well developed theory which is not presented here due to space limitations. It is hoped that the example at the end of the paper illustrates some of the main points.

The study of hidden modes and their classification as uncontrollable and unobservable leads to better understanding of the phenomena which occur in control design. For example, it can be shown that the hidden modes which can result in undesirable "ringing" of the input u in digital control (an example is given in [9]) are exactly those unobservable modes from y which are observable from u. Understanding the hidden modes leads to strategies on how to avoid them and in general it leads to better, more efficient control designs.

EXAMPLE

Consider the plant

$$P = \frac{s-1}{(s-2)(s+1)} \tag{16}$$

and suppose that a compensator should be designed to stabilize the plant and to attain

$$T = \frac{s-1}{(s+\alpha)(s+\beta)}. \tag{17}$$

One compensator that meets these specifications is

$$C = \left[\frac{-8(s+1)}{(s-13/3)}, \quad \frac{(s+1)^2(s+2/3)}{(s-13/3)(s+\alpha)(s+\beta)}\right]. \tag{18}$$

The closed-loop characteristic polynomial, $(s+\alpha)(s+\beta)(s+1)^2(s+2/3)$, is Hurwitz and of fifth order. Then it is clear that there are three stable hidden modes corresponding to the eigenvalues $\{-1,-1,-2/3\}$. By the discussion above these three hidden modes are uncontrollable from r and the pole of P at -1 corresponds to an unobservable mode from u.

Another controller C can be designed that will satisfy the specifications and will introduce less hidden modes. First, notice that no single degree of freedom controller can meet the specifications [14]. A simplified design approach for a two degrees of freedom controller is explained next. Since T is the same and T=NX=(s-1)X, X is the same. So we need to design C_y. We can do this by designing L. In order to minimize the number of hidden modes, the poles of L should be the same as the poles of X and no poles of L should be zeros of the plant. Then a possible choice for L is

$$L = \frac{\kappa}{(s+\alpha)(s+\beta)}, \tag{19}$$

where κ is a constant. The conditions for the compensator to be internally stabilizing and proper are X, L and $(1-LN)D^{-1}$ must be proper and stable. Only the latter one needs further checking:

$$(1-LN)D^{-1} = \frac{(s+\alpha)(s+\beta) - (s-1)\kappa}{(s-2)(s+1)(s+\alpha)(s+\beta)}. \tag{20}$$

The transfer function in (20) is stable if and only if $(2+\alpha)(2+\beta)=\kappa$, that is, $\alpha=(\kappa-4-2\beta)/(2+\beta)$. Then, for $\alpha>0$ need

$$0 < \beta < \frac{\kappa-4}{2} \tag{21}$$

or choose $\alpha>0$, $\beta>0$ and $\kappa=(2+\alpha)(2+\beta)$. By choosing κ in this way, a compensator $C=[-C_y \quad C_r]$ that attains the desired closed-loop transfer funtion is

$$C = \left[\frac{-\kappa(s-2)(s+1)}{(s+\alpha)(s+\beta)-(s-1)\kappa} \quad \frac{(s-2)(s+1)}{(s+\alpha)(s+\beta)-(s-1)\kappa} \right], \tag{22}$$

where there is a pole-zero cancellation at $s=-2$ in C_y and C_r. Suppose $\alpha=1$ and $\beta=3$, then $\kappa=15$ and

$$C = \left[-\frac{15(s+1)}{s-9} \quad \frac{s+1}{s-9} \right]. \tag{23}$$

For C in (23) there is only one hidden mode due to the interconnection of the controller and the plant; the pole of P at $s=-1$ corresponds to an uncontrollable mode from r and to an unobservable mode from u.

REFERENCES

[1] J. G. Truxal, Automatic Feedback Control System Synthesis, New York: McGraw Hill, 1955.

[2] I. M Horowitz, Synthesis of Feedback Systems, New York: Academic Press, 1963.

[3] P. J. Antsaklis, and M. K. Sain, "Unity Feedback Compensation of Unstable Plants," Proceedings of the 20th IEEE Conference on Decision and Control, pp. 305-308, December 1981.

[4] D. C. Youla and J. J. Bongiorno, "A Feedback Theory of Two-Degree-of-Freedom Optimal Wiener-Hopf Design," IEEE Transactions on Automatic Control, Vol. AC-30, pp.652-665, July 1985.

[5] G. Bengtsson, "Feedback Realizations in Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-22, pp. 576-585, August 1977

[6] L. Pernebo, "An Algebraic Theory for Design of Controllers for Linear Multivariable Systems - Parts I and II," IEEE Transactions on Automatic Control, Vol. AC-26, pp. 171-193, February 1981.

[7] C. A. Desoer and C. L. Gustafson, "Algebraic Theory of Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-29, pp.909-917, October 1984.

[8] M. Vidyasagar, Control System Synthesis: A Factorization Approach, Cambridge, Massachusetts: MIT Press, 1985.

[9] K. J. Åström and B. Wittenmark, Computer Controlled Systems. Theory and Design, Prentice-Hall, 1984.

[10] C. A. Desoer and A. N. Gündes, "Decoupling Linear Multiinput Multioutput Plants by Dynamic Output Feedback: An Algebraic Theory," IEEE Transactions on Automatic Control, Vol. AC-31, No. 8, pp. 744-750, 1986.

[11] T. Sugie and T. Yoshikawa, "General Solution of Robust Tracking Problem in Two-Degree-of-Freedom Control Systems," IEEE Transactions on Automatic Control, Vol. AC-31, pp.552-554, 1986.

[12] O. R. González and P. J. Antsaklis, "Existence and Characterization of Two Degrees of Freedom Stabilizing Controllers," Control Systems Technical Report No. 52, Department of Electrical and Computer Engineering, University of Notre Dame, November 1986.

[13] O. R. González and P. J. Antsaklis, "New Stability Theorems for the General Two Degrees of Freedom Control Systems," Proceedings of the 1987 International Symposium of Networks and Systems, June 1987.

[14] O. R. González, "Analysis and Synthesis of Two Degrees of Freedom Control Systems," Ph.D. Dissertation, University of Notre Dame, August 1987.

[15] P. J. Antsaklis and M. K. Sain, "Feedback Controller Parameterizations: Finite Hidden Modes and Causality," Multivariable Control, S. G. Tzafest (Ed.), Chapter 5, pp.85-104, Dordrecht Holland: D. Reidel, 1984.

[16] P. J. Antsaklis and M. K. Sain, "Feedback Synthesis with Two Degrees of Freedom: {G,H;P} Controller," Proceedings of the 9th World Congress of the International Federation of Automatic Control, Budapest, Hungary.

[17] F. M. Callier and C. A. Desoer, Multivariable Feedback Systems, New York: Springer-Verlag, 1982.

[18] O. R. González and P. J. Antsaklis, "Hidden Modes in Two Degrees of Freedom Design: {R;G,H} Controller," Proceedings of the 26th IEEE Conference on Decision and Control, pp. 703-704, 1986.

[19] D. C. Youla, H. Jabr and J. J. Bongiorno, Jr., "Modern Wiener-Hopf Design of Optimal Controllers -Part II: The Multivariable Case," IEEE Transactions on Automatic Control, Vol. AC-21, pp. 319-338, June 1976.

[20] G. Zames, "Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses," IEEE Transactions on Automatic Control, Vol. AC-26, pp. 301-320, April 1981.

[21] P. J. Antsaklis, "Some Relations Satisfied by Prime Polynomial Matrices and Their Role in Linear Multivariable System Theory," IEEE Transactions on Automatic Control, Vol. AC-24, pp. 611-616, August 1979.

[22] C. A. Desoer, R. W. Liu, J. Murray and R. Saeks, "Feedback System Design: The Fractional Representation Approach to Analysis and Synthesis," IEEE Transactions on Automatic Control, Vol.AC-25, pp. 399-412, June 1980.

[23] V. Kucera, Discrete Linear Control, New York: Wiley, 1979.

[24] B. Francis, J. W. Helton and G. Zames, "H^{∞}-Optimal Feedback Controllers for Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-29, pp. 888-900, October 1984.

[25] B.-C. Chang and J. B. Pearson, "Optimal Disturbance Reduction in Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-29, pp. 880-887, October 1984.

[26] J. C. Doyle, "Analysis of Feedback Systems with Structured Uncertainties," IEE Proceedings, Vol. 129, Part D, pp. 242-250, November 1982.

[27] ONR/Honeywell Workshop, Advances in Multivariable Control, Minneapolis, Minnesota, October 1984.

[28] J. W. Helton, "Worst Case Analysis in the Frequency Domain: The H^{∞} Approach to Control," *IEEE Transactions on Automatic Control*, Vol. AC-30, pp.1154-1170, December 1985.

[29] R. W. Liu and C. H. Sung, "On Well-Posed Feedback Systems in an Algebraic Setting," *Proceedings of the 19th IEEE Conference on Decision and Control*, pp. 269-271, December 1980.

[30] P. J. Antsaklis, "Feedback Systems," Lecture Notes, University of Notre Dame, Course EE 598C, Spring 1985.

[31] G. C. Newton, Jr., L. A. Gould and J. F. Kaiser, *Analytical Design of Linear Feedback Controls*, New York: Wiley, 1957.

[32] J. B. Cruz, Jr., Ed. *System Sensitivity Analysis*, Stroudsburg, Penn: Dowden, Hutchison and Ross, 1973.

[33] M. K. Sain, P. J. Antsaklis, B. F. Wyman, R. R. Geggi and J. L. Peczkowski, "The Total Synthesis Problem of Linear Multivariable Control - Part II: Unity Feedback and the Design Morphism," *Proceedings of the 20th IEEE Conference on Decision and Control*, pp. 875-884, December 1981.

[34] E. G. Gilbert, "Controllability and Observability in Multivariable Control Systems," *SIAM Journal of Control*, Vol. 2, No. 1, pp. 128-151, 1963.

[35] B. D. O. Anderson and M. R. Gevers, "On Multivariable Pole-Zero Cancellations and the Stability of Feedback Systems," *IEEE Transactions on Circuits and Systems*, Vol. CAS-28, pp.830-833, August 1981.

[36] C. A. Desoer and J. D. Schulman, "Cancellations in Multivariable Continuous-Time and Discrete-Time Feedback Systems Treated by Greatest Common Divisor Extraction," *IEEE Transactions on Automatic Control*, Vol. AC-18, pp.401-402, August 1973.

ON THE DISTANCE FROM AN UNCONTROLLABLE PAIR: A SUMMARY

MARK A. WICKS AND RAYMOND A. DECARLO
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

INTRODUCTION

A well accepted algorithm for computing the controllability of a linear time-invariant state model, the so-called staircase algorithm [4], provides only an upper bound on the perturbation required to make the model uncontrollable. A robust numerical algorithm should provide a reliable perturbation bound on its result so that one may determine whether any error introduced by numerical computation or any parameter uncertainty might result in a different answer. Eising considers this problem in [1] and [2]. Kenney and Laub expand on Eising's results in [3]. This summary describes an alternative formulation to the problem which yields a result similar to that of Eising.

MODEL DEFINITION

Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $x(\cdot) \in \mathbb{R}^{n}$. Define the linear time-invariant state model as follows:

$$\dot{x}(t)=Ax(t)+Bu(t). \tag{1}$$

PROBLEM STATEMENT

In [4], Paige suggests the measure, $\mu(A,B)$ as the norm of the smallest perturbation required to make the pair uncontrollable. In formulating the problem, let $U_i=\{(A,B)\in \mathbb{R}^{n\times n}\times \mathbb{R}^{n\times m} \mid rank[B \; AB \; \cdots \; A^{n-1}B]=n-i\}$, i.e., U_i is the set of all pairs whose controllable subspace has dimension i less than that of the entire state space. Let $U=\bigcup_{i=1}^{n} U_i$, i.e., U is the set of all uncontrollable pairs. The minimum distance problem can therefore be defined as

$$\mu(A,B) \triangleq \min_{\{(\hat{A},\hat{B})\in U\}} \|[A-\hat{A} \;\; B-\hat{B}]\|, \tag{2}$$

where $\| \cdot \|$ denotes the spectral norm of a matrix.

RESULTS

In [1] Eising shows that

$$\mu^2(A,B)=\min_{\substack{x\in C^n \\ \|x\|=1}} x^H A(I-xx^H)A^T x+x^H BB^T x, \tag{3}$$

where this holds for a restricted class of complex x's. In [2] he gives the perhaps more elegant formulation: $\mu(A,B)=\min_{\lambda\in C} \sigma_{min}[A-\lambda I \;\; B]$. Our formulation is based on the following result, given here without proof:

Theorem 1: $$\mu(A,B)= \min_{\{(\hat{A},\hat{B})\in U_1\cup U_2\}} \|[A-\hat{A} \;\; B-\hat{B}]\|.$$

The essence of the proof is showing that for any uncontrollable pair, there exists an arbitrarily small perturbation which increases the dimension of the controllable subspace by either one or two, but not necessarily both. Thus any pair from any of the U_i is arbitrarily close to a pair in either U_1 or U_2. The theorem implies that the problem is solvable by searching for the solution over the set, $U_1 \cup U_2$.

Proposition 1: The pair, $(\hat{A},\hat{B})$ is an element of $U_1 \cup U_2$ if and only if in some coordinate system, the pair has the representation,

$$\hat{A}=\begin{bmatrix}\hat{A}_{11} & \hat{A}_{12}\\ 0 & \hat{A}_{22}\end{bmatrix}\quad \hat{B}=\begin{bmatrix}\hat{B}_1\\ 0\end{bmatrix}. \tag{4}$$

where $\hat{A}_{22}$ either has dimension 2×2 or is a scalar.

Now for any orthogonal (norm preserving) coordinate transformation, $Q\tilde{x}=x$, of (1), it is possible to partition the transformed system as

$$\tilde{A}=\begin{bmatrix}\tilde{A}_{11} & \tilde{A}_{12}\\ \tilde{A}_{21} & \tilde{A}_{22}\end{bmatrix}\quad \tilde{B}=\begin{bmatrix}\tilde{B}_1\\ \tilde{B}_2\end{bmatrix}, \tag{5}$$

where $\tilde{A}_{21}$ has dimension $i\times(n-i)$ and $\tilde{B}_2$ has dimension $i\times m$. This leads to the following characterization:

Theorem 2: $$\min_{\{(\hat{A},\hat{B})\in U_i\}} |||[A-\hat{A}\ \ B-\hat{B}]||| = \min_{Q} |||[\tilde{A}_{21}\ \ \tilde{B}_2]|||.$$

Proposition 2: Let the orthogonal transformation, Q, be partitioned as $[Q_1\ Q_2]$, where Q_1 is $n\times(n-i)$ and Q_2 is $n\times i$. Then

$$|||[\tilde{A}_{21}\ \ \tilde{B}_2]||| = |||[Q_2^T A(I-Q_2Q_2^T)\ \ Q_2^T B]|||. \tag{6}$$

Derivation: First note that $[\tilde{A}_{21}\ \tilde{A}_{22}]=Q_2^T AQ$, $\tilde{A}_{22}=Q_2^T AQ_2$, and $\tilde{B}_2=Q_2^T B$. Clearly $|||[\tilde{A}_{21}\ \ \tilde{B}_2]|||^2=|||\tilde{A}_{21}\tilde{A}_{21}^T+\tilde{B}_2\tilde{B}_2^T|||$. From the above relationships,

$$\tilde{A}_{21}\tilde{A}_{21}^T=Q_2^T AQQ^T A^T Q_2^T-\tilde{A}_{22}\tilde{A}_{22}^T=Q_2^T AA^T Q_2-Q_2^T AQ_2Q_2^T A^T Q_2=Q_2^T A(I-Q_2Q_2^T)A^T Q_2.$$

Since the matrix, $(I-Q_2Q_2^T)$, is an orthogonal projector, we may write

$$(I-Q_2Q_2^T)=(I-Q_2Q_2^T)(I-Q_2Q_2^T)^T.$$

Finally, combining the above one obtains

$$|||[\tilde{A}_{21}\ \ \tilde{B}_2]|||^2=Q_2^T[A(I-Q_2Q_2^T)\ \ B][A(I-Q_2Q_2^T)\ \ B]^T Q_2.$$

Therefore,

$$|||[\tilde{A}_{21}\ \ \tilde{B}_2]|||=|||[Q_2^T A(I-Q_2Q_2^T)\ \ Q_2^T B]|||.$$

Note that the first $n-i$ columns of Q have no effect on the norms of interest. The problem in Theorem 1 is solved by performing the minimization of Theorem 2 using (6).

REFERENCES

[1] R. Eising, "The Distance Between a System and the Set of Uncontrollable Systems," *Proc. MTNS* (June 1983), Beer-Sheva, Israel, P.A. Fuhrmann (Ed.), Springer-Verlag, pp. 303-314, 1984.

[2] R. Eising, "Between Controllable and Uncontrollable," *Syst. Contr. Lett.*, vol. 4, pp. 263-264, 1984.

[3] C. Kenney and A.J. Laub, "Controllability and Stability Radii for Companion Form Systems," Apr. 1987, submitted to *Mathematics of Control, Signals and Systems* (Springer-Verlag).

[4] C.C. Paige, "Properties of Numerical Algorithms Related to Computing Controllability", *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 130-138, 1981.

MINIMUM EFFORT STATE FEEDBACK POLE-PLACEMENT FOR LINEAR MULTIVARIABLE SYSTEMS

N. F. Al-MUTHAIRI   S. P. BINGULAC
Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg , VA 24061

Abstract: A procedure to compute a state feedback matrix with minimal norm for the exact pole-placement in linear multivariable systems is described. The method is based on a direct transformation of the system matrix to a modified version of Luenberger's canonical forms.

INTRODUCTION

In nth-order multi-input , multi-output (MIMO) linear time-invariant system with m-inputs where exact pole assignment is required by state feedback , there are $n \times (m - 1)$ degrees of freedom available to satisfy additional performance criteria . In the literature , several methods were found to utilize this degree of freedom in designing state variable feedback with minimal norm. A unity rank controller was used in [1] where there are only $m - 1$ degrees of freedom available. In [2] an unrestricted rank state feedback matrix was found by minimizing a certain functional with respect to $n \times (m - 1)$ elements of the feedback matrix. However, for systems of order four or higher and since the functional in [2] is not smooth , the search for the minimum value of the functional with respect to all the variables becomes tedious and sometimes impossible. In [3] a successive method for pole shifting has been developed. In this approach the multilinear set of pole assignment equations is linearized by doing only small pole shiftings.

It was presented in [4] a modified canonical forms for linear multivariable systems . These realizations are modified version of Luenberger's canonical forms [5].

MODIFIED CANONICAL FORMS

Consider the following linear , time-invariant , multivariable system :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x \in R^n , \ u \in R^m \tag{1}$$

It is assumed that the system is controllable and $m \leq n$. Let the state feedback law is given by:

$$u(t) = Kx(t) \tag{2}$$

Then the closed-loop system will be :

$$\dot{x}(t) = (A + BK)x(t) \tag{3}$$

It is required to choose K with minimal norm from the family of $m \times n$ constant matrices which place the eigenvalues of system (3) at $\Lambda^d = \{\lambda_1^d, \dots, \lambda_n^d\}$.
The first step in the solution is to transform system (1) to a modified controllable canonical form. Let T be the transformation matrix, then the transformed system matrices will be:

$$\tilde{A} = TAT^{-1} \quad ; \quad \tilde{B} = TB \tag{4}$$

where

$$\tilde{A} = \left[\frac{\mathrm{O} \mid I_{n-m}}{A_{22}} \right] \; ; \; \tilde{B} = \left[\frac{\mathrm{O}}{B_2} \right] \tag{5}$$

The matrices A_{22} and B_2 are $m \times n$ and $m \times m$ arbitrary matrices respectively. Let $t^1, \dots, t^n$ be the rows of the matrix T , the algorithm to find them is:

$$\begin{aligned}
&t^{1} \; [B \mid AB \mid \cdots \mid A^{p-1}B] = 0 \\
&\;\vdots \\
&t^{k} \; [B \mid AB \mid \cdots \mid A^{p-1}B] = 0 \\
&t^{k+1}[B \mid AB \mid \cdots \mid A^{p-2}B] = 0 \\
&\;\vdots \\
&t^{m} \; [B \mid AB \mid \cdots \mid A^{p-2}B] = 0 \\
&t^{m+i_1} = t^{i_1}A \qquad i_1 = 1,2, \ldots ,min\{m,n-m\} \\
&\;\vdots \\
&t^{lm+i_l} = t^{(l-1)m+i_l}A \qquad i_l = 1,2, \ldots ,min\{m,n-lm\}
\end{aligned} \tag{6}$$

$$\text{where} \quad p = \binom{n}{m} , \quad k = n - mp , \quad 0 \le k \le m-1 \quad and \quad l = \binom{n-1}{m} \tag{7}$$

The symbol $\binom{a}{b}$ denotes the greatest integer value less than or equal to the ratio a/b .

Remark 1: The canonical form considered in (5) and the algorithm in (6) are for the generic case when the first n columns of the controllability matrix are linearly independent. However, for lack of space other cases will not be considered here.

Remark 2: The matrix T has some degrees of freedom by tuning a certain scalar(s).

MINIMUM EFFORT POLE-PLACEMENT ALGORITHM

Step 1: Plot the norm of T and the norm of the generalized inverse of $\bar{B}$ versus the free scalar. Then choose a particular value of the scalar when either of the above norms levels off.

Step 2: Find $\bar{A}$ and $\bar{B}$ using (4).

Step 3: Choose a matrix A_d which is of the same structure as $\bar{A}$ by minimizing :

$$J = \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij}^{22d} - a_{ij}^{22})^2 \tag{8}$$

subject to the constraints:

$$f_k(a_{ij}^{22d}) = c_k \qquad k = 0, \ldots ,n-1; \; i = 1, \ldots ,m; \; j = 1, \ldots ,n \tag{9}$$

where a_{ij}^{22} is the $(i,j)th$ element of the submatrix A_{22} , a_{ij}^{22d} is the $(i,j)th$ element of the submatrix A_{22d} in A_d which corresponds to A_{22} in $\bar{A}$, $f_k(.)$ is a function of the elements of A_{22d} and c_k's are the coefficients of the desired characteristics polynomial.

Step 4: Finally

$$K = \bar{B}^{+}(A_d - \bar{A})T \tag{10}$$

where $\bar{B}^{+}$ is the generalized inverse of $\bar{B}$.

Remark 3 : Since A_d has a special structure, there exists an algorithm to find $f_k(.)$.

Preliminary works indicate that it is easy to handle the minimization problem in step 3. Moreover, working with some examples, show clearly that the results obtained by using this method are better than those obtained in [2] and [3].

References:

1. Seraji, H.: Design of Multivariable Systems Using Unity-Rank State Feedback: Further Results. Electr. Lett. 11 (1975) pp. 34-35 .
2. Ramar, K. and Gourishankar, V.: Utilization of The Design Freedom of Pole Assignment Feedback Controllers of Unrestricted Rank. Int. J. Control 24 (1976) pp. 423-430.
3. Preuss, H. P.: Successive Pole Shifting by State Controllers of Prescribed Structure. Int. J. Control 33 (1981) pp. 1059-1071 .
4. Bingulac, S. P. , Padilla, R. A. and Padilla, C. S.: Modified Canonical Forms for Linear Multivariable Systems. Automatika 22 (1981), pp. 83-88 .
5. Luenberger, D. G.: Canonical Forms for Linear Multivariable Systems. IEEE Trans. Automat. Contr. 12 (1967) pp. 290-293 .

MEASURES OF CONTROLLABILITY AND OBSERVABILITY AND RESIDUES

D. K. LINDNER, J. BABENDREIER, and A. M. A. HAMDAN
Virginia Polytechnic Institute and State University
The Bradley Department of Electrical Engineering
Blacksburg, VA 24061

In this paper we consider the multiple input multiple output system given by

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned} \qquad (1)$$

where $x \varepsilon R^n = \mathcal{X}$, $y \varepsilon R^{\ell}$, *and* $u \varepsilon R^m$. The transfer function of (1) is given by

$$y(s) = G(s)\,u(s) = \left[\sum_{i=1}^{n} \frac{R_i}{(s-\lambda_i)} \right] u(s) \qquad (2)$$

if the poles of G(s) are distinct. Define the sets $\{p_i\}$ *and* $\{q_i\}$ as the sets of right and left eigenvectors. The $\ell x m$ residue matrix is given by

$$R_i = C p_i q_i^T B. \qquad (3)$$

In this paper we will develop the relationship between two geometric measures of controllability and observability and the residue matrices suggested by the formula (3). These measures are closely related to angles between subspaces of $\mathcal{X}$. Let $\mathcal{U}$ and $\mathcal{V}$ be subspaces of $\mathbb{C}^n$ with basis U and V respectively. We denote the canonical angles [2] by

$$\Theta(\mathcal{U},\mathcal{V}) = \max\ \{\theta_i\}. \qquad (4)$$

Let $c_j \triangleq$ *jth row of C*, $j = 1,\ldots,\ell$ and $b_k \triangleq$ *kth column of B*, $k = 1,\ldots,m$.

Definition 1 [1] For each right eigenvector p_i, we say that mode is ε_{oij} - modal unobservable in the jth output if

$$\cos\Theta(c_j^T, p_i) \le \varepsilon_{oij}. \qquad \square$$

Definition 2 [1] For each left eigenvector q_i we say that mode is ε_{oki} - modal uncontrollable from the kth intput if

$$\cos\Theta(q_i, b_k) \le \varepsilon_{oki}. \qquad \square$$

Define the diagonal matrices $D_o \triangleq diag\,(\|c_1\|, .., \|c_\ell\|)$, and $D_c \triangleq diag\,(\|b_1\|, .., \|b_m\|)$. Also let

$$\overline{C}p_i = \begin{bmatrix} \cos\Theta\,(c_1^T, p_i) \\ \vdots \\ \cos\Theta\,(c_\ell^T, p_i) \end{bmatrix}, \ \textit{and}\ \ q_i^T\overline{B} = \begin{bmatrix} \cos\Theta\,(q_i, b_1) \\ \vdots \\ \cos\Theta\,(q_i, b_m) \end{bmatrix}^T. \qquad (5)$$

Finally, let $abs\ (A) = [\,|a_{ij}|\,]$.

Lemma 3 [3]

$$abs\,(R_i) = \|p_i\|\ \|q_i\|\ D_o \overline{C} p_i q_i^T \overline{B}\ D_c \qquad \square$$

Next we introduce another measure of observability and controllability based on the following subspaces.

Definition 4 [2] The ith unobservable subspace $\mathcal{L}_i$ is defined by $\mathcal{L}_i \triangleq \bigcap_{j=0}^{i-1} N\,[CA^j]$, $\mathcal{L}_o \triangleq \mathcal{X}$.

Definition 5 [2] The ith reachable subspace $\mathcal{R}_i$ is defined by $\mathcal{R}_i \overset{\Delta}{=} \sum_{j=0}^{i-1} A^j B$, $\mathcal{R}_0 \overset{\Delta}{=} 0$.

We measure the distance between subspaces as follows:

Definition 6 [2] Let $\mathcal{U}$ and $\mathcal{V}$ subspaces of $\mathcal{C}^n$. The gap between $\mathcal{U}$ and $\mathcal{V}$ is the number

$$\tau(\mathcal{U},\mathcal{V}) = \max\{\sup_{\substack{\|u\|=1, v\varepsilon\mathcal{V} \\ u\varepsilon\mathcal{U}}}\inf \|v-u\|, \quad \sup_{\substack{\|v\|=1, u\varepsilon\mathcal{U} \\ v\varepsilon\mathcal{V}}}\inf \|v-u\|\}. \qquad \square$$

Remark 7 If $d(\mathcal{U}) = d(\mathcal{V})$ we can interpret the gap in terms of canonical angles [2], i.e.

$$\tau(\mathcal{U},\mathcal{V}) = |\sin \Theta(\mathcal{U},\mathcal{V})|. \qquad \square(3.1)$$

Definition 8 [2] Let $\mathcal{V}$ be an A-invariant subspace of (1). Then $\mathcal{V}$ is said to be ε_o -unobservable if there exists some j such that

$$\tau(\mathcal{V},\mathcal{L}_j) < \varepsilon_o. \qquad \square$$

Definition 9 Let $\mathcal{V}$ be an A-invariant subspace of (1). Then $\mathcal{V}$ is said to be ε_o -unocontrollable if there exists some j such that

$$\tau(\mathcal{V},\mathcal{R}_j) \le \varepsilon_o. \qquad \square$$

The following theorem relates the modal measures to the measures in Def. 8 and 9.

Theorem 10 Let $\mathcal{V}_{oi}$ and $\mathcal{V}_{ci}$ be subspaces of $\mathcal{C}^n$ such that $p_i \ \varepsilon\ \mathcal{V}_{oi}$ and $q_i \perp \mathcal{V}_{ci}$. Then

$$\cos \Theta(c_j^T, p_i) \le \tau(\mathcal{V}_{oi}, \mathcal{L}_h) \quad and \quad \cos \Theta(q_i, b_k) \le \tau(\mathcal{V}_{ci}, \mathcal{R}_h)$$

for every h = 1,..,n; i = 1,..,n; j = 1,..,ℓ, and k = 1,..,m.

Combining Theorem 10 with Lemma 3 we obtain a bound on the residue.

Theorem 11 Let $\mathcal{V}_{oi}$ and $\mathcal{V}_{ci}$ be subspaces of $\mathcal{C}^n$ such that $p_i \ \varepsilon\ \mathcal{V}_{oi}$ and $q_i \perp \mathcal{V}_{ci}$. Then for every $h_1, h_2 = 1,...,n$,

$$\|R_i\| \le \sqrt{\ell m}\ \|p_i\|\ \|q_i\|\ \tau(\mathcal{V}_{oi}, \mathcal{L}_{h_1})\ \tau(\mathcal{V}_{ci}, \mathcal{R}_{h_2}).$$

References

1. Hamdan, A.M.A. and Nayfeh, A., Modal Observability and Controllability Measures for First and Second Order Linear Systems and Model Reduction, preceedings of the Sixth VPI&SU/AIAA Symposium on Dynamics and Control of Large Structures, Dynamics and Control of Large Structures, Blacksburg VA, (June 1987).

2. D. K. Lindner, and W. R. Perkins, The Generalized Hessenberg Representation, Near Aggregation, and Near Unobservability, Decision and Control Report DC-91, Coordinated Science Laboratory, University of Illinois, Urbana IL (1987), also to appear in Automatica.

OBSERVER POLE ASSIGNMENT USING EIGENVALUE SENSITIVITY

M.A. CHOUDHRY AND C.S. SIMS
West Virginia University
Department of Electrical and Computer Engineering
Morgantown, WV 26506-6101

ABSTRACT

An expression is developed for the sensitivity of the eigenvalues of the observer with respect to the gain matrix. This expression may be used to assign the eigenvalues of observer for a multiple-input-output systems.

INTRODUCTION

Most control designs are based on full state feedback. All the states are rarely available in most of the practical systems. However, all the states can be reconstructed by a dynamic system called an observer [1]. The response of the observer depends upon the location of its eigenvalues. A method is presented in [2] to find the gain matrix for assigning the eigenvalues of the observer for multiple-input-output system. However, it lacks the flexibility of assigning eigenvalues associated to some states of particular interest. It is shown in [3] that there exists an association between groups of state variables and groups of natural modes of the system matrix. Eigenvalue sensitivity may be used to assign the eigenvalues associated with the states of particular interest.

PROBLEM FORMATION

Consider a linear time-invariant system described by the following state-space representation:

$$\dot{x}(t) = A\,x(t) + B\,u(t) \tag{1}$$

$$y(t) = C\,x(t)$$

Where A, B, and C are respectively, nxn, nxr, and (mxn) real constant matrices.

An observer for the above system can be written as:

$$\dot{\hat{x}}(t) = [A - K\,C]\,\hat{x}(t) + B\,u(t) + K\,y(t) \tag{2}$$

The poles of the observer are given by the characteristic values of the matrix $(A - KC = A_F)$. The error,

$$e(t) = x(t) - \hat{x}(t)$$

satisfies the differential equation

$$\dot{e}(t) = A_F\,e(t) \tag{3}$$

Let $\lambda_1\ \lambda_2\ , \ldots \lambda_n$ be the distinct eigenvalues and $\{ V_1\ V_2\ , \ldots V_n\}$ be the corresponding right eigenvectors of A_F;

Let $W = V^{-1} = [\, W_1\ W_2, \ldots W_n \,]$ be left eigenvectors of A_F.

The sensitivity of an eigenvalue λ_m of matrix A_F with respect to an element k_{ij} of gain matrix K is given by the following equations:

$$\frac{\partial \lambda_m}{\partial k_{ij}} = W_m^T \frac{\partial A_F}{\partial k_{ij}} V_m \tag{4}$$

Let K_i be the i-th row of gain matrix K.
Then

$$\frac{\partial \lambda_m}{\partial K_i} = \left\lfloor \frac{\partial \lambda_m}{\partial k_{i1}}, \frac{\partial \lambda_m}{\partial k_{i2}} \; . \; . \; . \; , \frac{\partial \lambda_m}{\partial k_{im}} \right\rfloor$$

$$= - W_{mi} V_m^T C^T \tag{5}$$

$$\frac{\partial \lambda_m}{\partial K} = - W_m V_m^T C^T \tag{6}$$

where $\frac{\partial \lambda_m}{\partial K}$ is an (n x m) matrix.

The values of the gain matrix K are obtained using an iterative procedure based on the sensitivities of the eigenvalue. The process is initiated with arbritrary matrix K such that the eigenvalues of the matrix (A - K C) are distinct. The change in dominant eigenvalue vector λ by $\Delta \lambda$ can be expressas follows:

$$\Delta \lambda = S \Delta K \tag{7}$$

$$\Delta K = S^{-1} \Delta \lambda \quad \text{and } K = K^0 + \Delta K \tag{8}$$

where K^0 is the gain matrix from the previous iteration.

Since the problem is nonlinear, the desired change $\Delta \lambda_{id} = \lambda_{id} - \lambda_i$ maynot be achieved in one step. Equations (6), (7) and (8) are solved iteratively until (Real λ_{id}) $\leqslant \varepsilon$ where ε is a small positive number. Thisalgorithm may not converge for all the cases.

CONCLUSION

An expression for the sensitivity of the observer poles with respect to gain matrix is developed. The proposed method gives design flexibility to assign dominant eigenvalues to some states of the observer.

ACKNOWLEDGEMENTS

The work reported in this paper was supported by the Office of Naval Research under Contract No. N00014-86-K-0651.

REFERENCES

[1] D. G. Luenberger, "Observers for multivariable systems," *IEEE Trans. on Automatic Control,* AC-11, pp. 190-197, April 1966.

[2] W. L. Brogan, "Applications of a determinant identity to pole-placement and observer problems," *IEEE Trans. on Automatic Control,* Vol. AC-19, pp. 612-614, October 1974.

[3] I. J. Perez - Arriaga, G. C. Verghese, and F. C. Schweppe, "Selective modal analysis with application to electric power systems. Part I: heuristic introduction," *IEEE Trans. on Power Apparatus,* Vol. PAS - 101, pp. 3117-3125, September 1982.

WORST-CASE ANALYSIS OF SPREAD-SPECTRUM SYSTEMS WITH OUTPUT QUANTIZATION

M.V. HEGDE
Electrical and Computer Engineering Department
Louisiana State University
Baton Rouge, LA 70803

W.E. STARK and D. TENEKETZIS
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109-2122

ABSTRACT

We model the process of communicating in the presence of interference, which is unknown or hostile, as a two-person zero sum game with the communicator and the jammer as the players. The objective function we consider is the mutual information. The communicator's strategies are distributions on the input alphabet and on a set of quantizers. The jammer's strategies are distributions on the noise power subject to certain constraints. We consider various conditions on the jammer's strategy set and on the communicator's knowledge. For the case with the decoder uninformed of the actual quantizer chosen we show that, from the communicator's perspective, the worst-case jamming strategy is a distribution concentrated at a finite number of points thereby converting a functional optimization problem into a non-linear programming problem. Moreover, we are able to characterize the worst-case distributions by means of necessary and sufficient conditions which are easy to verify. For the case with the decoder informed of the actual quantizer chosen we are able to demonstrate the existence of saddle-point strategies. The analysis is also seen to be valid for a number of situations where the jammer is adaptive.

1. INTRODUCTION

The applicability of game-theoretic models in jamming situations is by now well established [Root 61], [McEl 83], [Star 82], [Chan 85], [Peng 86]. In this paper we formulate fairly general models for a number of jamming situations as two-person zero-sum games between the communicator and the jammer. We allow the jammer the choice of one of a set of noise distributions satisfying peak and average power constraints. By way of counter-measure the communicator is allowed to randomize the input symbols as well as randomize the quantizer at the output side. We intend the analysis to be applicable to the performance of soft decision decoding for jammed channels.

As has been well known in the communication field, hard decision decoding loses roughly 2 dB in signal-to-noise ratio compared to soft decision decoding. Thus considerable interest has focused on soft-decision decoding. One problem that has been observed is that if a (soft) decoding algorithm designed for a nonjammed channel is used for a jammed channel then the performance is extremely poor when the jamming strategy is optimized. One method for "overcoming" this difficulty is to assume the jamming noise is one of two distributions (usually one having zero variance called the "off" state and the other called the "on" state) and that the decoder knows when the jammer is "on" and when the jammer is "off". Using this side information, similar results to the hard decision case have been obtained for the soft decision case [Simo 85] (for small rates there is no loss in performance). However, assuming this information may not be reasonable in a jamming situation. Most systems analyses do not incorporate jamming strategies that affect the reliability of the side information.

Thus there has been considerable interest over the last few years on decoding algorithms that do not assume side information and do not do hard decision decoding. However, most of these algorithms still assume the jammer pulses between one of two levels. In this paper we investigate the case of a decoder that processes symbols from a finite alphabet and where the only constraints on the jammer are average and peak power. We formulate the problem as a game with two players. The jammer whose strategy set consists of distributions on the power of the jamming noise, and the communicator, whose strategy set consists of a pair of distributions, one on the input alphabet and one on the set of quantizers. We look for worst-case jamming strategies and investigate when the game admits of a saddle point. We do the analysis using mutual information as our objective function.

The paper is organized as follows. In Section 2 we define the models we will be considering and give examples for which our models apply. In Sections 3 and 4 we state our results concerning the worst case jamming strategy and the optimal quantizer strategy for the cases with decoder uninformed about the actual quantizer chosen and with decoder informed about the actual quantizer chosen respectively. Finally in Section 5 we discuss our results and state our conclusions and extensions.

2. CHANNEL MODELS

In this section we describe the models we use in the subsequent analysis. In all cases we consider a modulator that transmits one out of M signals in D dimensions ($D \leq M$). This transmitted signal is denoted by the random variable X. The received signal which is corrupted by the jammer in some fashion is demodulated and quantized into one of L values. The received signal is denoted by the random variable Y. (Y can also be a random vector without changing any of the following analysis).

The general philosophy that we will use is that of game theory with the players being the jammer and the communicator. The jamming strategies are distributions dF on D random variables, $Z_1, Z_2, ..., Z_D$. These random variables represent the power of the jammer in each of the signal dimensions and are modelled as modulating a generic noise variable present in the channel. For example, if $D = 1$ and N is a zero-mean, variance 1 Gaussian random variable then the jammer's noise may be of the form $Z_1 N$. The jammer, however, has an average power constraint and a peak power constraint. More generally the jammer is constrained by

$$\int f(z_1, z_2, ..., z_D) dF(z_1, z_2, ..., z_D) \leq K_J \tag{1}$$

and

$$0 \leq Z_j \leq b_j \;\; j = 1, \ldots, D \tag{2}$$

where b_j is the peak power constraint and $f(z_1, ..., z_D)$ is some continuous functional of $(z_1, ..., z_D)$. For average power constrained channels with no peak constraint we let b_j become very large.

The output of the demodulator is quantized into one of L values, say $0, 1, ..., L-1$. The output of the quantizer, Y, is also the output of the channel for coding. The strategies for the communicator are to choose a distribution, $dG(\theta)$, on the quantization thresholds and a distribution, $dP(x)$, on the input alphabet. We will let Q parametrize the quantizers and assume Q is some compact subset of R. The random quantizer will be denoted by Θ and θ a particular realization. For each $(z_1, ..., z_D)$ and θ there is a probability distribution on the output of the channel given the input of the channel:

$$Prob\{Y = y|X = x, \Theta = \theta, Z_1 = z_1, Z_2 = z_2, ..., Z_D = z_D\} = p(y|x, \theta, z_1, z_2, ..., z_D). \tag{3}$$

The above model describes the input output relation of the channel for a particular symbol. In addition we model the channel as being memoryless.

We now introduce some notation. Let:
$A = \{0, 1, ..., M-1\}$ be the input alphabet,
$B = \{0, 1, ..., L-1\}$ be the output alphabet,
Θ be the quantizer parameter space (some compact subset of R)
Z be $(Z_1, ..., Z_D)$, $\quad (0 \leq Z_i \leq b_i)$
$p(y|x, \theta, z)$, the transition probability from x to y given θ, z, and
$P_{yx}(\theta, z)$ the corresponding stochastic matrix, $P_{yx}(\theta, z) = [p(y|x, \theta, z)]$.

We assume that
(i) $p(y|x, \theta, z)$ is continuous in z for all θ, x and
(ii) $p(y|x, \theta, z)$ is continuous in θ for all x, z.

Let **S** denote the set of all probability distributions on the Borel sets of $K \triangleq \{\underline{z} = (z_1, \ldots, z_D) : 0 \leq z_i \leq b_i\}$, and

$$\begin{aligned} I(G, P; F) &= I\left(\int_K \int_Q P_{yx}(\theta, z) dG(\theta) dF(z)\right) \\ &= I\left(\int_K P_{yx}^G(z) dF(z)\right) \\ &= I\left(\int_Q P_{yx}^F(\theta) dG(\theta)\right) \\ &= I\left(\overline{P}_{yx}(G, F)\right) \end{aligned} \tag{4}$$

where $I\left(\overline{P}_{yx}(G, F)\right)$ is the mutual information whenever X and Y are related by the stochastic matrix $\overline{P}_{yx}$.

The performance measure we are interested in is the largest rate such that nearly error-free communication can be achieved, i.e. channel capacity. Another performance of interest is the channel cutoff rate, R_0, since many researchers believe this to be a practical limit to the set of rates for which reliable communication is possible. Similar results to those in this paper can be derived with R_0 as the performance measure (see [Hegd 87]).

We consider two different information structures for the communicator:

I. The decoder is unaware of the actual quantizer chosen but only knows the distribution $dG(\theta)$ on the set of quantizers. The jammer knows only the set of quantizers but not the distribution $dG(\theta)$ chosen by the communicator. He is also aware that the decoder does not know the actual quantizer chosen.

II. The decoder knows the actual quantizer chosen. Again the jammer knows only the the set of quantizers. He also knows that the decoder is aware of the actual quantizer chosen.

Case I is seen to apply to situations where, for reasons of implementation perhaps, the decoding is fixed and not altered with the specific quantizer chosen. It may also be viewed as worst-case in the sense that the decoder's knowledge of the specific quantizer and the utilization of such knowledge can only improve the communicator's performance. When there is no randomization of the quantizer, i.e. the quantizer is fixed, Cases I and II are the same and our results for both cases apply to that situation.

Communicator's Perspective

The communicator is interested in the maximum rate at which information can be reliably transmitted no matter what strategy the jammer employs. The communicator designs his system assuming the jammer will somehow find out the strategy he is using and then choose the worst possible distribution on the power levels. In Case I the largest rate for which information can reliably be transmitted is

$$\max_{G,P} \min_{F} I(G,P;F)$$

where $I(G,P;F) \triangleq I(X;Y)$ when (dG,dP) is chosen by the communicator and dF is chosen by the jammer. That this is the maximum rate of reliable transmission is well known since what we are dealing with is a compound channel with a finite input alphabet and a finite output alphabet [Csis 81, pgs. 172-173].

Jammer's Perspective

The jammer is interested in the minimum rate such that information can not be reliably transmitted at any higher rate no matter what strategy the communicator employs. The jammer designs his system assuming the communicator will somehow find out the strategy he is using and then design the optimal communication system. In Case I the smallest rate that the jammer can guarantee reliable communication can not be above is

$$\min_{dF} \max_{dG,dP} I(G,P;F).$$

That this is the smallest rate the jammer can guarantee is obvious since for each F the rate above which reliable communication is impossible is $\max\limits_{dG,dP} I(G,P;F)$. In case II the appropriate mutual information can be written as an expectation of the mutual information for a fixed θ:

$$I(G,P;F) = E_G(I(\theta,P;F))$$

where E_G refers to taking expectations w.r.t. dG and $I(\theta,P;F) \triangleq I(X;Y|\theta)$.

In all of our analysis we assume that the jammer and the decoder/quantizer have complete information about the set of strategies possible for each other so that no secret information is considered. As mentioned previously, the performance measure we consider is the largest rate such that reliable communication (in the sense of arbitrarily small error probability) is possible. The type of channels we are considering are known as compound channels. We consider the strategies (distributions) by the jammer to be constant for a whole codeword as opposed to (possibly) changing after each symbol of a codeword which would correspond to an arbitrarily varying channel. For compound channels the capacity with finite input and output is well known to be the maximum of the minimum mutual information. The minimum is over all possible transition probabilities and the maximum is over all probability distributions on the input to the channel. Thus, using the maximum of the minimum mutual information as the performance measure corresponds to the largest rate such that reliable communication is possible no matter what strategy the jammer employs. We are now ready to state the results. In brief our results show that when the decoder is informed of the quantization rule then (under a compatibility assumption), there is a saddlepoint in cases A and B, i.e. the jammer's rate and the communicator's rate are equal (Theorem 4). When the decoder is not informed of the quantization rule then the jammer's rate and the communicator's rate may differ. However, the optimal distributions, F from the communicator's point of view and G from the jammer's point of view are finite dimensional (in all the cases A, B, C and D) (Theorem 1). This converts a functional optimization problem into a finite-dimensional non-linear programming problem.

3. CASE AI: DECODER UNINFORMED

The communicator has to determine the distributions $(dG(\theta), dP(x))$ that maximize the amount of information $I(G,P;F)$ transmitted. The jammer has to find the noise distribution $dF(z)$ to minimize the information received by the decoder. Thus, the communicator's goal is to achieve

$$\max_{dG(\theta),dP(x)} \ \min_{dF(z)} I(G,P;F)$$

whereas the jammer wants to achieve

$$\min_{dF(z)} \ \max_{dG(\theta),dP(x)} I(G,P;F).$$

In this section we show that for any choice of strategy of either player there is a simple characterization of the optimal reaction strategy of his opponent.

Theorem 1: a) The jammer can achieve the minimum in $\max\limits_{dG(\theta),\ dP(x)} \ \min\limits_{dF(z)} \ I(G,P;F)$ with a distribution concentrated at at most $M(L-1)+2$ points.

b) The communicator can achieve the maximum in $\min\limits_{dF(z)} \ \max\limits_{dG(\theta),\ dP(x)} \ I(G,P;F)$ with a distribution concentrated at at most $M(L-1)+1$ points.

Discussion: Theorem 1(a) says that the communicator in trying to achieve $\max\limits_{dG(\theta),\ dP(x)} \ \min\limits_{dF(z)} \ I(G,P;F)$ has to consider only reaction strategies of the jammer that have a finite number of points of support, i.e. for each $(dG(\theta), dP(x))$ chosen by the communicator the worst-case jammer distribution may be assumed to be concentrated at a finite number of points and this number is bounded uniformly (in $(dG(\theta), dP(x))$) by $M(L-1)+2$. It follows that for a fixed quantizer (i.e. no randomization of the quantization) the worst-case jammer is one who chooses such a finite-dimensional distribution. Similarly Theorem 1(b) says that the jammer may, from his perspective of trying to achieve $\min\limits_{dF(z)} \ \max\limits_{dG(\theta),\ dP(x)} \ I(G,P;F)$, consider only finite dimensional reaction strategies on the communicator's part.

Proof of Theorem 1: See [Hegd 87].

For channels which are symmetric for each θ and z i.e. $p(y|x_1,z,\theta)$ is some permutation of $p(y|x_i,z,\theta)$ part(a) of the theorem holds with $(L-1)+2 = L+1$ instead of $M(L-1)+2$ and part (b) of the theorem holds with L instead of $M(L-1)+1$. For M-ary symmetric channels, i.e. channels with M inputs and M outputs and such that for each θ and z, $p(y_i|x_i,z,\theta) = 1-\epsilon$ and $p(y_i|x_j,z,\theta) = \dfrac{\epsilon}{M-1}$, $i \neq j$, the bound on the number of points of support reduces to 3 for part (a) and to 2 for part (b).

Necessary and Sufficient Conditions

We now characterize the aforementioned finite-dimensional distributions by means of necessary and sufficient conditions. Given any $dG(\theta)$ and the power constraint we define

$$U_c(K_J,G) \triangleq \sup_{\substack{F \in S \\ h_F \leq K_J}} -I(G,P;F) \tag{5}$$

where $h_F \triangleq \int_K f(z)\, dF(z)$. To simplify notation we define

$$D : \mathbf{S} \rightarrow \mathbf{R} \text{ by } D(F) = \int_K f(z) dF(z) - K_J. \tag{6}$$

We formulate necessary and sufficient conditions for the characterization of the optimal distributions of Theorem 1 in the following two theorems.

Theorem 2: $U_c(G, K_J)$ is achieved by a distribution $F_0 \in \mathbf{S}$ satisfying $D(F) \leq 0$ and a necessary and sufficient condition for $U_c(G, K_J) = -I(G, P; F_0)$ is that for some constant $\lambda \geq 0$

$$\int_K [-i(z; G, F_0) - \lambda f(z)] dF(z) \leq -I(G, P; F_0) - \lambda K_J \tag{7}$$

for all $F \in \mathbf{S}$

where $i(z; G, F_0) \triangleq \sum_{x,y} p(x)\, p(y|x,z) \log \left(\dfrac{\int p(y|x,z)\, dF_0(z)}{\sum_x p(x) \int p(y|x,z)\, dF_0(z)} \right)$.

Proof of Theorem 2: See [Hegd 87].

From Theorem 1 we know that it is possible to find F_0 from the set of distributions with a finite number of points of support. Finding such an F_0 entails determining the set of points of increase as well as the amounts of increase of F_0 at those points. Let E_0 denote the set of points of increase of F_0. We now state

Theorem 3: Let F_0 be a probability distribution satisfying the power constraint. Then F_0 achieves $U_c(G, K_J)$ iff for some $\lambda \geq 0$

C1) $-i(z; G, F_0) \leq -I(G, P; F_0) + \lambda(f(z) - K_J)$

for all $z \in K$.

C2) $-i(z; G, F_0) = -I(G, P; F_0) + \lambda(f(z) - K_J)$

for all $z \in E_0$.

Proof of Theorem 3: See [Hegd 87].

Theorems 1 and 3 reduce the calculation of the distributions describing the reaction strategies to finite-dimensional non-linear programming problems. They can be used to simplify the search for conservative strategies which are optimal for either player.

4. CASE AII: DECODER INFORMED

We have an arbitrary joint distribution on $Z_1, \ldots, Z_D$. The jammer chooses $dF(z)$ and knows that the decoder knows Θ. The communicator chooses $dG(\theta)$ and further the decoder knows θ.

In this case we make a "compatibility" assumption, that is, for every θ and $dF(z)$ the capacity-achieving input distribution $dP(x)$ remains the same.

While "compatibility" certainly restricts our model applicability, we show by example that it is often a worst-case assumption. For instance, we know [Dobr 59] that if $M = L$ and if the jammer's strategy set is restricted such that for each distribution $dF(z)$ and quantizer θ, Prob $\{\text{error}|x\} \leq \epsilon$ for every x, then the saddle-point strategy for the jammer is to choose a distribution such that

$$p(y|x) = \frac{1}{M} \quad \text{for all } y,x \text{ if } \epsilon > 1 - \tfrac{1}{M}$$

and

$$\begin{aligned} p(y|x) &= \frac{\epsilon}{M-1} \quad y \neq x \text{ if } \epsilon \leq 1 - \tfrac{1}{M} \\ &= 1-\epsilon \quad y = x \end{aligned}$$

and the saddle-point strategy for the communicator is to choose a uniform distribution on the input alphabet. In our model this corresponds to choosing the canonical noise variables so that $p(y|x,\theta)$ is a symmetric channel for each θ. Such symmetry (and thereby "compatibility") is obtained in a number of other situations as a saddle-point strategy. Under certain conditions, when we have convex constraints in the M noise variables affecting the M inputs of the channel which are invariant under any permutation of the M variables (i.e. a "symmetric" constraint) then the choice of a uniform distribution on the input and the choice of a symmetric channel are saddle-point strategies for the communicator and the jammer respectively (see [Hegd 87], Appendix E). To describe one more example, if we have M inputs and M outputs,

$$\begin{aligned} y_i &= n_i \quad i = 1, \ldots, M \;\; i \neq j \\ y_j &= A + n_j \quad i = j, \end{aligned}$$

n_i are $N(0, v_i), i = 1, \ldots, M$ independent random variables and there is further the constraint $\sum_{i=1}^{M} v_i = c$, then, from arguments similar to those in [Hegd 87, Appendix E], it can be seen that the saddle point strategy is to choose $v_i = \frac{c}{M}$ and a uniform distribution on the input.

Utilization of the "compatibility" assumption allows us to write the objective functions as

$$\min_{dF(z)} \max_{dG(\theta)} E_G(C(\theta, F)).$$

and

$$\max_{dG(\theta)} \min_{dF(z)} E_G(C(\theta, F))$$

where $C(\theta, F) = \max_{dP(x)} I(\theta; F)$ and $I(\theta; F) = I(X; Y|\theta)$.

The main result is stated in the following theorem:

Theorem 4: There exists a pair of distributions $dG^*(\theta), dF^*(z))$ such that

$$E_G(C(\theta, F^*)) \leq E_{G^*}(C(\theta, F^*)) \leq E_{G^*}(C(\theta, F))$$

for all feasible $dG(\theta), dF(z)$, i.e., $(dG^*(\theta), dF^*(z))$ is a saddle point for the game in case AII.

Proof of Theorem 4: See [Hegd 87].

We note that these saddle-point distributions need not have finite support. However, in this case we have an equilibrium and with no further knowledge of each other's choice of strategy, the jammer and the quantizer should be content utilizing $dG^*(\theta)$ and $dF^*(z)$. We can also derive necessary and sufficient conditions at these saddle points as in Theorem 2.

Fixed Quantizer

Before concluding this section we also point out that if we did not have randomized quantization then without "compatibility" the game would have a saddle- point where the jammer's saddle-point distribution need be concentrated at at most $M(L-1)+2$ points. We summarize this in Theorem 5.

Theorem 5: For any quantizer θ, there exists a pair of distributions $dP^*(x), dF^*(z)$ such that

$$I(\theta,P,F^*) \le I(\theta,P^*,F^*) \le I(\theta,P^*,F) \tag{8}$$

for all feasible dP, dF. Moreover, $dF^*(z)$ can be chosen to be concentrated at at most $M(L-1)+2$ points. Necessary and sufficient conditions for $dF^*(z)$ and $dP^*(x)$ are:

For some $\lambda_1, \lambda_2 \ge 0$

$$-i(z;\theta,F^*) \le -I(\theta,P^*,F^*) + \lambda_1(f(z) - K_J) \tag{9}$$

for all $z \in K$ and

$$-i(z;\theta,F^*) = -I(\theta,P^*,F^*) + \lambda_1(f(z) - K_J) \tag{10}$$

for all $z \in E_0$ where $i(.;.,.)$ is as defined in Theorem 2 with G concentrated on θ. Also

$$I_x(\theta,P^*,F^*) = \lambda_2 \tag{11}$$

for all $x \ni P^*(x) > 0$ and

$$I_x(\theta,P^*,F^*) \le \lambda_2 \tag{12}$$

for all $x \ni P^*(x) = 0$, where

$$I_x(\theta,P^*,F^*) \triangleq \sum_y p(y|x,\theta) \log \frac{p(y|x,\theta)}{\sum_x P^*(x)p(y|x,\theta)}.$$

Proof of Theorem 5: See [Hegd 87].

5. CONCLUSIONS

We have constructed fairly general channel models which are capable of representing a number of jamming situations. The jammers considered have all been non-adaptive. Using results from the compound channel we are able to give operational significance to our minimax performance measures, i.e. to assert the existence of encoders and decoders which can perform at arbitrarily low probabilities of error at rates close to our performance measures. Our analysis is also clearly applicable to many restrictions on the jammer's strategy set other than the ones we have considered.

In the case of the decoder informed we have shown that one need only search among the class of distributions with finite support to determine a worst-case strategy for the jammer and communicator. We have found a bound on the number of these points of support in terms of the sizes of the input and the output alphabet. Thus, we have reduced the computation of the worst case jamming strategies to a finite-dimensional non-linear programming problem. Moreover, we have been able to characterize these distributions by necessary and sufficient conditions which are fairly easy to test.

With the decoder informed we reduce the communicator's strategy set either by using the "compatibility" assumption or by fixing a quantizer. In this case we have demonstrated the existence of saddle-point strategies, and have further characterized these strategies when the quantizer is not randomized.

As we have mentioned earlier all the above is for non-adaptive jamming. Because of our objective function, the compound channel is the appropriate model for interpretation of the results. We can allow for more sophisticated jammers if we allow the jammer's strategies to depend on the previous (and present) channel inputs. The appropriate channel model to use then is that of the arbitrarily "star" varying channel (A^*VC) [Csis 81, pg.233]. This model generalizes the arbitrarily varying channel (AVC) and includes it as a special case. It is known that the m-capacity (i.e. capacity with maximum probability of error over all the codewords) of the A^*VC is the same as that of the corresponding AVC [Csis 81, pg.232]. For binary output alphabet and finite input alphabet the capacity is known to be equal to $\max\limits_{dP(x)} \min\limits_{W \in \overline{\overline{\mathcal{W}}}} I(X;Y)$ where X and Y are the input and the output respectively, W is any channel chosen from the set of channels $\mathcal{W}$ and $\overline{\overline{\mathcal{W}}}$ is the row-convex closure of $\mathcal{W}$ [Csis 81, Section 2.6]. In our case the set of channels is already row-convex closed, hence the appropriate objective functions would be:
a) For the communicator:

$$\max_{(dG(\theta),dP(x))} \min_{dF(z)} I(G,F) .$$

b) For the jammer

$$\min_{dF(z)} \max_{(dG(\theta),dP(x))} I(G,F) .$$

These are the same objective functions used in Section 3. We also obtain the same objective functions as in Section 4 when the decoder informed. Thus, all the results derived in Sections 3 and 4 can be extended to the A^*VC channel with binary output. The m-capacity of the AVC is as yet unknown for output sizes greater than 2. On the other hand the a-capacity of the AVC (i.e. the capacity with average probability of error) is known to be either 0 or else $\max\limits_{dP(x)} \min\limits_{W \in \overline{\mathcal{W}}} I(X;Y)$ where $\overline{\mathcal{W}}$ is the convex closure of the set $\mathcal{W}$ to which W belongs [Csis 81, pg.214]. Since in our model the set of channels is convex as well as row-convex the a-capacity is known to be greater than 0 iff the m-capacity is greater than 0 [Ahls 78]. Thus with average probability of error whenever the jammer's strategy set is such that he cannot force the capacity to be 0 then all the results of Sections 3 and 4 extend to the case of the A^*VC channel.

ACKNOWLEDGEMENT

The work of M.V. Hegde and W.E. Stark was supported by the Office of Naval Research under contract N00014 85 K05645. The work of D. Teneketzis was supported by the National Science Foundation under Grant no. ECS-8517708 and by a Rackham Research Grant of the University of Michigan.

REFERENCES

[Ahls 78] R. Ahlswede, "Elimination of correlation in random codes for arbitrarily varying channels", *Zeitschrift fur Wahrscheinlichkeitstheorie*, no. 33, pp. 159-175, 1978

[Aubi 82] J.P. Aubin, *Mathematical Methods of Game and Economic Theory*, North-Holland, 1982

[Chan 85] L.F. Chang, An Information-Theoretic Study of Ratio-Threshold Antijam Techniques, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1985

[Csis 81] I. Csiszar and J. Korner, *Information Theory : Coding Theory for Discrete Memoryless Systems*, Academic Press, 1981

[Dobr 59] R.L. Dobrushin,"Optimum information transmission through a channel with unknown parameters", *Radio Engineering Electronics*, vol. 4, no. 12, 1959

[Dubi 62] L.E. Dubins,"On extreme points of convex sets", Journal of Mathematical Analysis and Applications, pp. 237-244, 1962

[Hegd 87] M.V. Hegde, Performance Analysis of Coded, Frequency-Hopped Spread-Spectrum Systems, Ph.D. Thesis, University of Michigan, Aug. 1987

[McEl 83] R.J. McEliece,"Communication in the presence of jamming- an information theoretic approach", in *Secure Digital Communications*, Springer- Verlag, pp. 127-166 1983

[Peng 86] W.C. Peng, Some Communication Jamming Games, Ph.D. Thesis, University of Southern California, Jan. 1986

[Root 61] W.L. Root,"Communication through unspecified additive noise", *Information and Control*, vol. 4, pp. 15-29, 1961

[Simo 85] M.K. Simon, J.K. Omura, R.A. Scholz, B.K. Levitt, *Spread Spectrum Communications, vols 1,2 and 3*, Computer Science Press, 1985

[Smit 71] J.G. Smith,"The information capacity of amplitude- and variance- constrained scalar gaussian channels", *Information and Control*, vol. 18, pp.203-219, 1971

[Star 82] W.E. Stark, Coding for Frequency-Hopped Spread-Spectrum Channels with Partial-band Interference, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1982

[Star 86] W.E. Stark, D. Teneketzis, and S.K. Park,"Worst-case analysis of partial-band interference", *Proceedings of the 1986 Conference on Information Sciences and Systems* , 1986

[Wits 80] H.S. Witsenhausen," Some aspects of convexity useful in information theory', *IEEE Transactions on Information Theory*, vol. 26, pp.265-271, May 1980

DYNAMIC CLIPPING FOR DETECTION OF FH/BFSK SIGNALS WITH DIVERSITY

Y. K. KIM and C. C. LEE
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60201

ABSTRACT

The detection of frequency hopped binary frequency shift keying signals with diversity under partial band interference is studied. In this paper, we propose two kinds of dynamic clipping techniques. We analyze the performance for the case of two channels of diversity and compare the results to those of the fixed clipping receiver and the linear combining receiver. The results show that both the proposed systems have jamming-suppression capabilities without requiring side information about jamming. And it is shown that one of the proposed system performs nearly as well as the fixed clipping receiver and that the other has very small performance degradation under jamming-free condition.

I. INTRODUCTION

Spread Spectrum (SS) communication system is a system in which the transmission signal is spread over a much wider frequency band than the bandwidth normally required for the information being sent. There are several advantages of SS systems over conventional communication systems, such as message hiding, interference rejection and selective addressing capability [12],[13],[14].

Along with the SS system, diversity transmission also has been known as an effective way of communication over severely faded or jammed channel. Therefore, combination of SS system with diversity transmission techniques can provide powerful communication method under hostile environments.

In this paper, we consider non-coherent detection of Frequency Hopping Binary Frequency Shift Keying (FH/BFSK) signal with L-channel diversity. The major topic in the receiving of an FH signals with diversity is the strategy as to how to combine observations from diversity channels. Under jamming-free conditions as well as full-band jamming conditions, linear combining receiver which originates from likelihood ratio test is known as the optimum receiver [1],[5]. However, performance of the linear combining receiver degrades substantially under partial-band jamming conditions. To improve the system performance under partial-band interference, several non-linear combining schemes have been proposed [2],[3],[4],[11],[15].

It is our objectives to find systems which can effectively suppress partial-band interference without any side information while maintaining a performance comparable to that of linear combining receiver under interference-free condition.

The original idea of our system came from the consideration that the reliability of one channel can be better understood by referring to the output of other channels.

In the proposed systems, we apply the idea stated above to the clipping technique which performs well once the optimum clipping threshold is determined. The proposed systems use averages of observed "mark" and "space" signals from all diversity channels as a clipping threshold for "mark" and "space" channels respectively.

The organization of this paper is as follows. In Section II we discuss jamming strategies briefly and define analysis model of jamming signal. In Sections III and IV, we define analysis models for Dynamic Clipper and Weighted Average Dynamic Clipper and obtain the probability density function (p.d.f) of decision variable for L=2. In Section V, numerical results of both systems are obtained for L=2 and compared to the existing results.

II. About Jamming Strategy

Since a spread spectrum(SS) system utilizes an extremely wide bandwidth for information transmission, a jammer needs a huge power to jam the entire bandwidth. Therefore, a jammer usually concentrates its jamming power on a randomly selected part of frequency band the SS system use.

We assume that we do not have any knowledge on jamming condition except that jamming bandwidth is wide enough to cover at least a single hopping frequency cell. That is, once the "space"(mark) channel of a particular diversity channel is jammed, the "mark"(space) channel is also jammed. For the purpose of analysis, we assume that a total jamming power of J watts is uniformly spread over B_J Hertz and employ a Gaussian model for it. Then the probability γ that a particular hopping frequency cell is jammed is given by

$$\gamma = \frac{B_J}{W} \tag{2-1}$$

where W is the bandwidth of the SS system.

If we assume that the bandwidth of each hopping cell is B, the jamming power in the jammed cell becomes

$$\sigma_J^2 = \left(\frac{B}{\gamma \cdot W}\right) J , \qquad 0 < \gamma < 1 \tag{2-2}$$

During our system analysis, we define signal to jamming ratio(SJR) as

$$SJR = 10\log\left(\frac{E_s}{N_J}\right) \tag{2-3}$$

where E_s is total signal power and

$$N_J = \left(\frac{J}{W}\right).$$

We shall analyze the system performance for various γ and SJR.

III. Dynamic Clipping Receiver

Fig. 1 shows a block diagram of the proposed system. We assume that the bandwidth of each diversity channel is B_d Hz and the data rate R bits/sec.

The transmitter selects one of two baseband frequencies f_1 and f_2 according to incoming data bit. Then the selected signal of frequency f_i, i=0,1 is passed through a BPF of bandwidth B Hz, where B equals R. Then a set of RF frequencies are selected for diversity channels according to a Pseudo-Noise(PN) sequence generator's output. Then the filtered baseband signal modulates the selected RF frequencies and each modulated signal is transmitted through appropriate diversity channel.

At the receiver, the signal from each diversity channel is demodulated using frequencies assigned by receiver PN sequence generators. Assuming perfect synchronization between receiver and transmitter PN sequence generators, the demodulated output $r_i(t)$ from the ith diversity channel will have the following form[1]:

$$r_i(t) = s_i(t) + n_i(t) \qquad i=1,2,\ldots,L \tag{3-1}$$

where

$$s_i(t) = \begin{cases} \sqrt{2S}\,\cos(2\pi f_1 t + \alpha_i), & \text{when "space" was sent} \\ \sqrt{2S}\,\cos(2\pi f_2 t + \beta_i), & \text{when "mark" was sent} \end{cases}$$

$$n_i(t) = \begin{cases} n_o(t) + n_J(t), & \text{when jamming is present} \\ n_o(t), & \text{otherwise} \end{cases}$$

where S:signal power per bit

f_1:baseband frequency for "space"

f_2:baseband frequency for "mark"

α_i,β_i:phase angles each of which is uniformly distributed in $(0,2\pi)$

n_o:Gaussian noise with variance σ_o^2.

Assume that "space" was sent, the output of quadratic detector at the i^{th} channel can be written as

$$r_{i1}(t) = (\sqrt{2S}\cos\alpha_i + x_i(t))^2 + (\sqrt{2S}\sin\alpha_i + y_i(t))^2 \tag{3-2}$$

and

$$r_{i2}(t) = x_i(t)^2 + y_i(t)^2 \tag{3-3}$$

where $x_i(t), y_i(t)$ are quadrature and in-phase components of noise $n_i(t)$, respectively and $r_{i1}(t), r_{i2}(t)$ are space and mark signals, respectively. Then $r_{i1}(t)$ and $r_{i2}(t)$ have noncentral chi-squared distribution and chi-squared distribution function, respectively[6].

In the dynamic clipping receiver, the decision rule can be obtained as follows:

$$Z_n(t) = \sum_{i=1}^{L} \min\,[r_{in}(t) + g_n(t)] \tag{3-4}$$

where n = 1, 2 index space and mark respectively, and

$g_n(t) = \sum_{i=1}^{L} \frac{1}{L} r_{in}$ are thresholds for "space" and "mark" respectively.

The final decision variable Z can be written as

$$Z(t) = Z_2(t) - Z_1(t) \tag{3-5}$$

and the decision rule:

$$Z(t) \underset{\text{space}}{\overset{\text{mark}}{\gtrless}} 0. \tag{3-6}$$

For L=2 case, let

$$Z_n^1(t) = r_{1n}(t) + g_n(t), \qquad \text{if } r_{1n} < r_{2n} \tag{3-7}$$

$$Z_n^2(t) = r_{2n}(t) + g_n(t), \qquad \text{if } r_{1n} > r_{2n} \tag{3-8}$$

The above equations can be written as follows:

$$Z_n^1(t) = \frac{3}{2} r_{1n}(t) + \frac{1}{2} r_{2n}(t), \qquad 0 \leq r_{1n} < r_{2n}$$

$$= r'_{1n}(t) + r'_{2n}(t) \tag{3-9}$$

Assume that space was sent, the p.d.f. of $Z_1^1(t)$ and $Z_2^1(t)$ can be obtained as follow. Note that

$$r'_{21}(t) = Z_1^1(t) - r'_{11}(t), \qquad 0 \leq r'_{21}(t) < \frac{3}{4} Z_1^1(t) \tag{3-10}$$

The Jacobian of this transformation is unity. The resulting p.d.f. becomes

$$P_{Z_1^1}(p) = \int_0^{\frac{3}{4}p} \frac{1}{3\sigma_1^2\sigma_2^2} \exp[-\frac{\lambda_1}{2} - \frac{\lambda_2}{2} - \frac{p}{3\sigma_1^2} - \frac{(p-r'_{11})}{\sigma_2^2}] \cdot$$

$$I_o[(\frac{\lambda_1}{3\sigma_1^2})^{\frac{1}{2}}] I_o[(\frac{\lambda_2(p-r'_{11})}{\sigma_2^2})^{\frac{1}{2}}] \, dr'_{11} \tag{3-11}$$

where $\lambda_i = \frac{2S}{\sigma_i^2}$ and $I_o(x)$ is the modified Bessel function of zero order.

Similarly, we have

$$P_{Z_2^1}(p) = \frac{1}{3\sigma_1^2 - \sigma_2^2} \exp[(-\frac{p}{4\sigma_1^2} - \frac{p}{4\sigma_2^2}) - \exp(-\frac{p}{\sigma_2^2})] \tag{3-12}$$

Finally, note that the p.d.f. of $Z_1^2(t)$ and $Z_2^2(t)$ can be obtained by interchanging σ_1 and σ_2 in Eq.(3-11) and Eq.(3-12), respectively. Using Eq.(3-6), the p.d.f. of decision variable Z(t) becomes

$$P_Z(p) = \int_0^\infty P_{Z_1}(q) P_{Z_2}(p+q) \, dq \tag{3-13}$$

where $P_{Z_1}(p) = P_{Z_1^1}(p) + P_{Z_1^2}(p)$ and $P_{Z_2}(p) = P_{Z_2^1}(p) + P_{Z_2^2}(p)$

The probability of error can be written as

$$P(\varepsilon|\sigma_1\sigma_2) = \int_0^\infty P_Z(p) \, dp \tag{3-14}$$

$$= \int_0^{\infty}\int_0^{\infty} P_{Z_1}(q)P_{Z_2}(p+q)\, dqdp$$

Changing the order of integrations and carrying out the inner integration yields

$$P(\varepsilon|\sigma_1\sigma_2) = 2\int_0^{\infty} P_{Z_1}(p)\cdot$$

$$\{\frac{1}{3\sigma_2^2-\sigma_1^2}[\frac{4\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}\exp(-\frac{p}{4\sigma_1^2}-\frac{p}{4\sigma_2^2}) - \sigma_1^2\exp(-\frac{p}{\sigma_1^2})]$$

$$+\frac{1}{3\sigma_1^2-\sigma_2^2}[\frac{4\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}\exp(-\frac{p}{4\sigma_1^2}-\frac{p}{4\sigma_2^2}) -\sigma_2^2\exp(-\frac{p}{\sigma_2^2})]\}dp \qquad (3\text{-}15)$$

The probability of error under partial-band jamming condition is given by

$$P_J(e) = \sum_{m=0}^{L} \binom{L}{1}\gamma^m(1-\gamma)^{L-1}P(e|m) \qquad (3\text{-}16)$$

where $P(e|m)$ denotes conditional probability of error, given that n of L channels are jammed. From Eq.(3-19) and Eq.(3-20), we obtain the following expression for the Probability of error under jamming:

$$P(e)=\gamma^2P(\varepsilon|\sigma_t\sigma_t) + 2\gamma(1-\gamma)[P(\varepsilon|\sigma_t\sigma_o) + P(\varepsilon|\sigma_o\sigma_t)] + (1-\gamma)^2P(\varepsilon|\sigma_o\sigma_o) \qquad (3\text{-}17)$$

where $\sigma_t^2 = \sigma_o^2 + \sigma_J^2$.

Eq.(3-17) can be solved using numerical integration[8],[9]. During the analysis, we vary γ between 1 to 10^{-3} for given SNR and SJR, to find the worstcase partial-band jamming condition. Also we analyze the performance of fixed clipping receiver for comparison. In the analysis of the fixed clipping receiver, optimum threshold for given SNR has been used.

IV. Weighted Average Dynamic Clipping Receiver

The block diagram of this system is shown in Fig. 1. which is the same as the dynamic clipper case except for the averaging circuits. In this scheme, the clipping level g(t) for "space" and "mark" channels are decided by weighted average as follows:

$$g_{in}(t) = \frac{1}{2}[A_{in}(t)v_{in}(t) + B_{in}(t)r_{in}(t)] \qquad (4\text{-}1)$$

where

$$v_{in}(t) = \frac{1}{L-1}\sum_{\substack{k=1\\k\neq i}}^{L} r_{kn}$$

$$B_{in}(t) = \begin{cases} \exp\left[- \dfrac{\dfrac{r_{in}(t)}{v_{in}(t)} - 1}{C}\right], & \text{when } r_{in}(t) > v_{in}(t) \\ 1, & \text{when } r_{in}(t) \leq v_{in}(t) \end{cases}$$

$$A_{in}(t) = 2 - B_{in}(t)$$

and C is parameter to be optimized with respect to the system performance under jamming-free and partial-band jamming conditions.

It follows that

$$Z_n(t) = \sum_{i=1}^{L} \min\,[r_{in}(t) + g_{in}(t)] \tag{4-2}$$

We confine our performance analysis of the weighted average dynamic clipper to L=2 case because of the following reasons. First, the analysis becomes excessively sophisticated as L becomes large. Second, the efficiency of the dynamic clipping receiver discussed in the previous section increases as L becomes large. Therefore, the advantage of the weighted average dynamic clipping receiver decreases for larger L.

With L=2 and Eq.(3-7) we have

$$Z_n^1(t) = r_{1n}(t) + \frac{A_{2n}(t)r_{1n} - B_{2n}(t)r_{2n}}{2}, \quad \text{if } r_{1n} < r_{2n} \tag{4-3}$$

where $B_{2n}(t) = \exp\left[- \dfrac{\dfrac{r_{2n}(t)}{r_{1n}(t)} - 1}{C}\right]$ and $A_{2n}(t) = 2 - B_{2n}(t)$.

For the simplicity of analysis, we quantize r_{2n}/r_{1n} to N levels. Although both of these numbers have values between 0 and infinity, we can limit their maximum value to certain range where $B_{2n}(t)$ becomes negligibly small because of the exponential term involved in $B_{2n}(t)$. Then

$$B_{2nj}(t) = \exp\left[- \frac{C_{nj} - 1}{C}\right], \quad \text{when } C_{nj}r_{1n} < r_{2n} < C_{nj+1}r_{1n} \tag{4-4}$$

where $j=1,2,\ldots,N$

From now on, we drop the variable t and suscript 2 in the coefficients A and B for convenience. Then, we have

$$Z_n^1(t) = \frac{A_{nj} + 2}{2}\, r_{1n}(t) + \frac{B_{nj}}{2}\, r_{2n}(t)$$

$$= r'_{1n}(t) + r'_{2n}(t), \qquad \frac{B_{nj}C_{nj}}{2 + B_{nj}}\, r'_{1n} < r'_{2n} < \frac{B_{nj}C_{nj+1}}{2 + B_{nj}}\, r'_{1n} \tag{4-5}$$

If we let $r'_{2n}(t) = Z_n^1(t) - r'_{1n}(t)$, the Jacobian becomes 1 and $r'_{2n}(t)$ satisfies the following inequality

$$X_{nj}(t) < r'_{2n}(t) < Y_{nj}(t) \tag{4-6}$$

where $X_{nj}(t) = \dfrac{2 + A_{nj}}{2 + A_{nj} + B_{nj}C_{nj+1}} Z_n^1(t)$ and $Y_{nj}(t) = \dfrac{2 + A_{nj}}{2 + A_{nj} + B_{nj}C_{nj}} Z_n^1(t)$

Then the p.d.f. of $Z_1^1(t)$ can be written as

$$P_{Z_1^1}(p) = \sum_{j=1}^{N-1} \int_{X_{1j}}^{Y_{1j}} \frac{1}{(2 + A_{1j})B_{1j}} \exp\left[- \frac{\lambda_1}{2} - \frac{\lambda_2}{2} - \frac{r'_{11}}{(2 + A_{1j})\sigma_1^2} - \frac{(p - r'_{11})}{B\sigma_2^2}\right] I_o\left[\left(\frac{2\lambda_1 r'_{11}}{(2 + A_{1j})\sigma_1^2}\right)^{\frac{1}{2}}\right] I_o\left[\left(\frac{2\lambda_2(p - r_{11})}{B_{1j}\sigma_2^2}\right)\right]^{\frac{1}{2}} dr_{11}$$

$$+ \int_0^{Y_{1N}} \frac{1}{(2 + A_{1N})B_{1N}} \exp\left[- \frac{\lambda_1}{2} - \frac{\lambda_2}{2} - \frac{r'_{11}}{(2 + A_{1N})\sigma_1^2} - \frac{(p - r'_{11})}{B\sigma_2^2}\right] I_o\left[\left(\frac{2\lambda_1 r'_{11}}{(2 + A_{1N})\sigma_1^2}\right)^{\frac{1}{2}}\right] I_o\left[\left(\frac{2 \lambda_2(p - r_{11})}{B_{1N}\sigma_2^2}\right)\right]^{\frac{1}{2}} dr_{11} \quad (4\text{-}7)$$

Similarly, the p.d.f. of $Z_2^1(t)$:

$$P_{Z_2^1}(p) = \frac{1}{(2 - B_{2j})\sigma_1^2 - B_{2j}\sigma_2^2} \Big\{ \sum_{j=1}^{N-1} \Big[\exp\Big(- \frac{p}{2 + B_{2j} + B_{2j}C_{2j}} \Big(\frac{1}{\sigma_1^2} + \frac{C_{2j}}{\sigma_2^2}\Big)\Big) - \exp\Big(- \frac{p}{2 + B_{2j} + B_{2j}C_{2j+1}} \Big(\frac{1}{\sigma_1^2} + \frac{C_{2j+1}}{\sigma_2^2}\Big)\Big)\Big]$$

$$+ \exp\Big(- \frac{p}{2 + B_{2N} + B_{2N}C_{2N}} \Big(\frac{1}{\sigma_1^2} + \frac{C_{2N}}{\sigma_2^2}\Big)\Big) - \exp\Big(- \frac{p}{\sigma_2^2} \Big)\Big\} \quad (4\text{-}8)$$

As was in previous section, the p.d.f. of $Z_1^2(t)$ and $Z_2^2(t)$ can be obtained by interchanging σ_1 and σ_2 in Eq.(4-7) and Eq.(4-8) respectively. The p.d.f. of the decision variable Z(t) can be obtained similarly.

$$P_Z(p) = \int_0^{\infty} P_{Z_1}(q)P_{Z_2}(p+q)\, dq \quad (4\text{-}9)$$

where $P_{Z_1}(p) = P_{Z_1^1}(p) + P_{Z_1^2}(p)$

$P_{Z_2}(p) = P_{Z_2^1}(p) + P_{Z_2^2}(p).$

The probability of error of the weighted average dynamic clipping receiver for given noise variances σ_i can be obtained using Eq.(3-16) and Eq.(3-17).

In our analysis, we used a 7-level "logarithmically uniform" quantization with upper limit 723 on the value of r_{2n}/r_{1n}. The constant C was chosen to be the smallest value such that the resulting degradation in bit error rate(compared to linear combining receiver) under jamming-free condition is less than twenty percent.

V. Performance comparison and discussion

We compare the worst-case probability of error to those of the linear combining receiver, the fixed clipping receiver and the weighted average dynamic clipping receiver in Fig. 2, Fig. 3 and Fig. 4. As the results show, performance of the dynamic clipping receiver is worse than that of the optimum fixed clipping receiver. This is partly because efficiency of jamming suppression during averaging process becomes very poor for small L. For example, if output of one channel is very large due to the jamming, the maximum efficiency of the interference suppression can be expressed as (L-1)/L. Therefore, we need more efficient jamming suppression scheme especially for small L. Also Fig. 2, Fig. 3 and Fig. 4 show that the performance of the weighted average dynamic clipping receiver is compatible with that of the fixed clipper system, which has to be optimized with respect to system SNR, for both jamming-free and partial-band jamming conditions. We also plot the bit error probability as a function of SNR when there is no partial-band interference in Fig. 5. The results suggest that the proposed system improves system performance under the partial-band jamming conditions while degradation under jamming-free condition is negligible.

In this paper, we have concentrated our analysis on performance under partial-band jamming and considered only L=2 assuming a fading free and single access channel. To complete our study on the dynamic clipping techniques, we need to evaluate the system performance for larger L.

References

[1] J. M. Wozencraft, and I. M. Jacobs, Principles of communication Engineering. New York: Wiley, 1965.
[2] J. S. Lee, R. H. French, and L. E. Miller, "Probability of error analyses of a BFSK frequency-hopping system with diversity under partial-band jamming interference-Part I:Performance of square-law linear combining receiver," IEEE Trans. Communication., vol. COM-32, pp.654-653, June 1984
[3] J. S. Lee, L. E. Miller, and Y. K. Kim, "Probability of error analyses of a BFSK frequency-hopping system with diversity under partial-band jamming interference-Part II: Performance of square-law nonlinear combining soft decision receivers," IEEE Trans. Communication., vol. COM-32, pp.1243-1250, Dec. 1984.
[4] L. E. Miller, J. S. Lee, and A. P. Kadrichu, Probability of error analyses of a BFSK frequency-hopping system with diversity under partial-band jamming interference-Part III: Performance of square-law self normalizing soft decision receivers," IEEE Trans. Communication., vol. COM-34, pp.669-675, July 1985.
[5] N. F. Krasner, "Optimal detection of digitally modulated signals," IEEE Trans. Communication., vol. COM-30, pp.885-895, May 1982.
[6] A. D. Whalen, Detection of Signals in Noise. New York:Academic, 1971.
[7] G. N. Watson, A Treatise on the Theory of Bessel Functions. Cambridge: University Press, 1966.

[8] P. J. Davis and P. Rabinowitz, Numerical Integration. Waltham: Blaisdell, 1967.

[9] M. L. James, G. M. Smith, and J. C. Wolford, Applied Numerical Methods for Digital Computation With FORTRAN and CSMP. New York: Dun-Donnelley, 1977.

[10] A. J. Viterbi and J. K. Omura, Principles of Digital Communication and Coding. New York: McGraw Hill, 1979.

[11] P. J. Lee, "Performance of a normalized energy metric without jammer state information for an FH/MFSK system in worst case partial band jamming," IEEE Trans. Communication, vol. COM-33, pp.869-877, Aug. 1985.

[12] R. C. Dixon, Spread Spectrum System, New York: Wiley, 1976.

[13] D. J. Torrieri, Principles of Military Communication Systems, Dedham: Artech, 1981.

[14] J. K. Holmes, Coherent Spread Spectrum Systems, New York: Wiley, 1982.

[15] C. M. Keller and M.B. Purseley, "Diversity combining and viterbi's ratio threshold technique for frequency hop communications in the presence of partial-band interference," Proc. of 19th Conf. Information Sciences and Systems, The John Hopkins University, Baltimore, MD, pp.532-537, Mar. 1985.

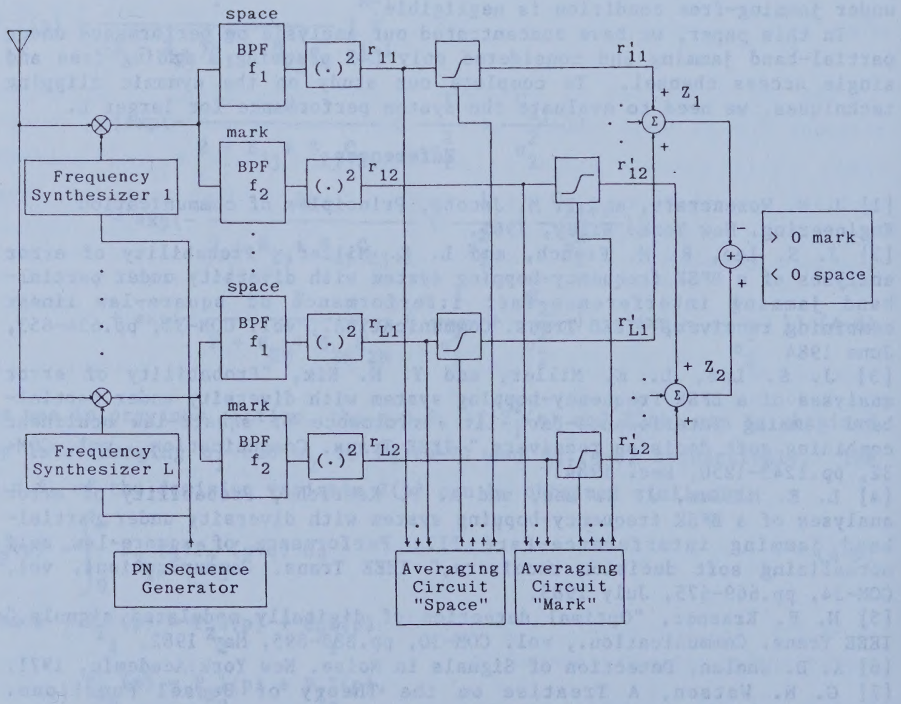

space
BPF f_1
$(.)^2$
r_{11}
r'_{11}
$+ Z_1$
Σ
+
mark
BPF f_2
$(.)^2$
r_{12}
r'_{12}
Frequency Synthesizer 1
−
+
> 0 mark
< 0 space
space
BPF f_1
$(.)^2$
r_{L1}
r'_{L1}
$+ Z_2$
Σ
+
mark
BPF f_2
$(.)^2$
r_{L2}
r'_{L2}
Frequency Synthesizer L
PN Sequence Generator
Averaging Circuit "Space"
Averaging Circuit "Mark"

Fig. 1. Block diagram of dynamic clipping receiver

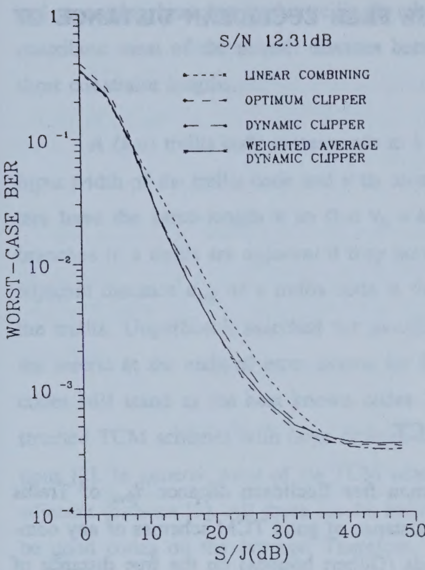


Fig. 2. Worst-case jamming performance of proposed systems for L=2 and SNR =12.31 dB

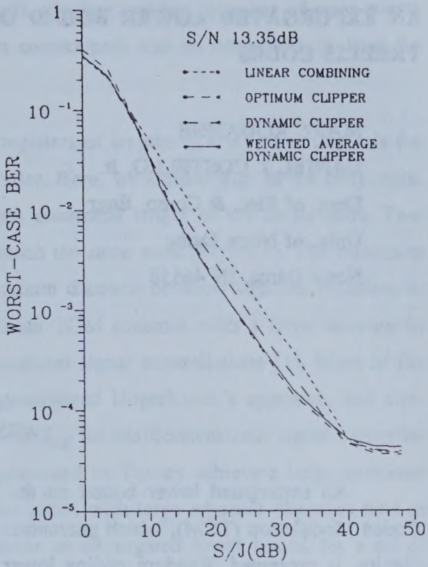


Fig. 3. Worst-case jamming performance of proposed systems for L=2 and SNR =13.35 dB

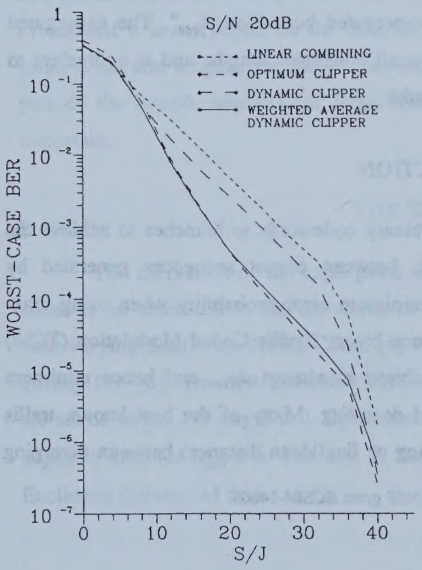


Fig. 4. Worst-case jamming performance of proposed systems for L=2 and SNR =20 dB

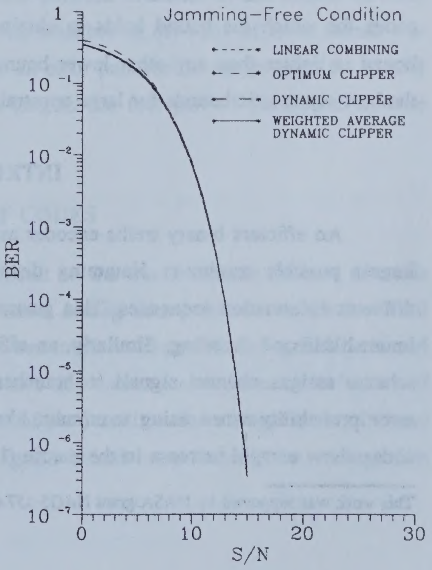


Fig. 5. Comparison of BER under jamming-free condition for L=2

AN EXPURGATED LOWER BOUND ON THE FREE EUCLIDEAN DISTANCE OF TRELLIS CODES

MARC ROUANNE
DANIEL J. COSTELLO, Jr.
Dept. of Elec. & Comp. Engr
Univ. of Notre Dame
Notre Dame, IN 46556

ABSTRACT

An expurgated lower bound on the minimum free Euclidean distance d_{free} of Trellis Coded Modulation (TCM), which guarantees the existence of good TCM schemes of any complexity, is presented. Random coding lower bounds (Gilbert bounds) on the free distance of binary codes in particular, and of trellis codes in general, are not very tight for small constraint lengths, whereas they are exponentially tight for large constraint lengths. We tighten the best lower bound known on the free Euclidean distance of TCM schemes by expurgating the set of codes for which the bound holds to obtain an "expurgated bound on d_{free}". The expurgated bound is tighter than any other lower bound for small constraint lengths and is equivalent to the best asymptotic bounds for large constraint lengths.

INTRODUCTION

An efficient binary trellis encoder assigns binary codewords to branches to achieve the largest possible minimum Hamming distance d_h between output sequences generated by different information sequences. This guarantees minimum error probability when using maximum likelihood decoding. Similarly, an efficient non-binary Trellis Coded Modulation (TCM) scheme assigns channel signals to branches to achieve maximum d_{free}, and hence minimum error probability when using maximum likelihood decoding. Many of the best known trellis codes show a rapid increase in the metric (Hamming or Euclidean distance) between diverging

---

This work was supported by NASA grant NAG5-557-02 and NSF grant ECS84-14608.

and remerging branches in the trellis. In other words, the first and last branches of error events contribute most of the overall distance between a correct path and an error path, at least for short constraint lengths.

A (k,ν) trellis code corresponds to k shift registers of lengths $\nu_i \leq \nu$ $(1 \leq i \leq k)$. k is the input width of the trellis code and ν its memory order. Here, we assume that all the shift registers have the same length ν so that $\nu_0 = k\nu$ is the constraint length of the trellis code. Two branches in a trellis are *adjacent* if they leave or reach the same state (Figure 1). The minimum adjacent distance d_{adj} of a trellis code is the minimum distance between adjacent branches in the trellis. Ungerboeck searched for pseudo-optimum TCM schemes with a large increase in the metric at the ends of error events for 2-dimensional signal constellations [1]. Most of his codes still stand as the best known codes. Wei generalized Ungerboeck's approach, and constructed TCM schemes with large or even maximum d_{adj} for multidimensional signal constellations [2]. In general, most of the TCM schemes presented by Forney achieve a large minimum adjacent distance [3]. All these results indicate that codes with large adjacent distances tend to be good codes on the average. Therefore, we derive an expurgated lower bound for a set of codes with large minimum adjacent distance.

The expurgated lower bound on the free Euclidean distance of trellis codes consists of two parts: a random coding bound on the distance between branches in the middle of error events and a lower bound on the distance between branches at the ends of error events. For small constraint lengths, the major contribution to an error path metric comes from the second part of the bound, whereas for large constraint lengths the first part of the bound is more important.

THE SET OF CODES

The derivation of the expurgated lower bound on the free Euclidean distance of trellis codes is an extension of the derivation of the random coding bound derived for the set of all time-varying Euclidean Trellis Codes [4] which owes much to Chernoff, Gallager, Viterbi and Forney [5] -[8]. The first difference between the two derivations comes from the definition of the set of codes. For any $\delta \geq 0$, C_δ denotes the set of (k,ν) trellis codes c which have minimum adjacent distance $d_{adj} \geq \delta$. For any δ, the expurgated bound lower bounds the average free Euclidean distance of codes in C_δ.

The following Lemmas answer the questions: Is it possible, for the same trellis code, to maximize the distance between diverging branches and the distance between remerging branches? How do these maximizations affect the distances between branches that are not adjacent? The Lemmas are based on the notion of 2-state *subtrellis*. For example, Figure 2b shows the two 2-state subtrellises of the 4-state trellis in Figure 2a.

Lemma 1: Given a signal constellation S which contains M equiprobable signals, Y_t and Y_t' represent two non-adjacent branches which correspond to the same time interval t, and y_t is the signal labeling Y_t, then the probability $p(y_t'|y_t)$ over all codes in C_δ that y_t' is the label for Y_t' is

(i) $p(y_t'|y_t) \leq M\, p(y_t')$ if Y_t and Y_t' belong to the same 2-state subtrellis,

(ii) $p(y_t'|y_t) = p(y_t')$ if Y_t and Y_t' do not belong to the same 2-state subtrellis.

Lemma 2 upper bounds the probability of a particular labeling of an error path given the labeling of the correct path over all codes in C_δ.

Lemma 2: Given a labeled path $\underline{y}$ in a (k, ν) trellis code in C_δ, the probability of a labeling $\underline{y}'$ for an error path of length τ is bounded by

$$p(\underline{y}'|\underline{y}) = \prod_{t=1}^{\tau} p(y_t'|y_t) \;\leq\; p(y_1'|y_1)\, p(y_\tau'|y_\tau)\, M^{2B(\tau,\nu)} \prod_{t=2}^{\tau-1} p(y_t'|y_t),$$

where $B(\tau,\nu) \triangleq \lceil \tau/\nu - 2 \rceil_+$ and $\lceil x \rceil_+$ is the smallest non-negative integer larger than or equal to x.

Proving Lemma 2 consists in counting the number of branches in an error event which satisfy (i) or (ii) and are not at the ends of the error event.

DERIVATION OF THE BOUND

The first step consists in stating what must be proved

Assertion: For any $\delta \geq 0$ and any expurgated set C_δ of codes c and for any $\alpha \geq 0$

$$e^{\alpha d^2} \sum_{c \in C_\delta} p(c) \sum_{\underline{y} \in c} \sum_{\underline{y}' \in e_{\underline{y}}(c)} e^{-\alpha d_e^2(\underline{y},\underline{y}')} < 1 \;\Rightarrow\; \exists\, c_0 \in C_\delta \text{ s.t. } d_{free}(c_0) > d,$$

where $e_{\underline{y}}(c)$ is the set of error paths in c diverging from $\underline{y}$ at time 1.

The next step, what Forney calls configuration counting, consists of regrouping the labeled paths $\underline{y}$ (or $\underline{y}'$) that correspond to the same unlabeled paths $\underline{Y}$ (or $\underline{Y}'$). This is very similar to configuration counting over the set C of all time varying trellis codes, since it essentially depends on the topology of the trellis. This step depends heavily on Lemma 2. The last step puts the bound in its final form:

Theorem: Given a signal constellation S and a distance δ, there exists a (k, ν) trellis code with maximum minimum adjacent distance $d_{adj} \geq \delta$ and minimum free Euclidean distance d_{free} such that:

$$d_{free}^2 > 2\delta^2 + \max_{\begin{cases} E(\alpha) > k\ln 2 + 1/\nu \ln M \\ 0 \leq \alpha \end{cases}} \left[(\nu - 2)\,\frac{E(\alpha)}{\alpha} - \frac{\theta'[E(\alpha)]}{\alpha} \right],$$

where

$$E(\alpha) \triangleq - \ln \sum_{y \in S} p(y) \sum_{y' \in S} p(y')\, e^{-\alpha d_e^2(y,y')},$$

$d_e(y,y')$ is the Euclidean distance between signals y and y',

$$\theta'[E(\alpha)] \triangleq \ln \frac{1}{e^{\eta} - 1} \quad \frac{1 - e^{-\nu\eta}}{1 - e^{-\nu\eta + \ln M}},$$

$\eta \triangleq [E(\alpha) - k\ln 2]$, and

α is a parameter which optimizes the bound.

APPLICATION OF THE BOUND

The expurgated bound is mostly useful for small constraint lengths, since it is equivalent to random coding bounds for large constraint lengths. For example, if the Euclidean distance can be made proportional to the Hamming distance as in BPSK modulation, then the bound applies to binary trellis codes like convolutional codes. Table 1 shows a comparison of the expurgated bound with the best random coding bound [9], the best known codes [10], and Heller's very tight upper bound [11], for rate 1/4 convolutional codes.

Since upper bounds are not as tight as the Heller bound for TCM schemes used with M-ary modulation, the bound is most useful for trellis codes used with M-ary modulation like M-PAM or M-PSK signal constellations. Figure 3 shows the expurgated bound for 8-PSK modulation compared with the best known rate 2/3 codes [1], upper bounds [12]-[13], and the random coding bound [4].

The expurgated bound, like any other random coding bound on convolutional or trellis codes, applies only to rate *k*/*n* codes generated by *k* shift registers of equal length. However, trellises that represent shift registers of various lengths can be decomposed into ***nested trellises*** [14], to which the bound applies. Figure 4 shows a general representation of nested trellises. Figure 5 shows a 2 level nested trellis (Figure 5c) where the branches of the trellis in Figure 5b have been replaced with the trellis in Figure 5b (the input width *k* determines the number of levels of a nested trellis). For example, Ungerboeck's trellises are nested trellises where parallel branches represent 1-state trellises (Figure 5d). When the distance between parallel branches limits the achievable free distance of a code, they can be replaced with trellises to introduce redundancy between sequences of signals from the same subsets (Figure 4) [15]. The free distance of a nested trellis code satisfies

$$d_{free} = \min_{1 \leq i \leq k} (d_{free}(\nu_i)),$$

where $d_{free}(\nu_i)$ is the free distance of any subtrellis of level i, where a subtrellis of level i represents the last $k - i + 1$ shift registers [14]. The expurgated bound on d_{free} can be used to lower bound each $d_{free}(\nu_i)$ by replacing the distances between signals with distances between subsets of signals that correspond to level i (Figure 4). This gives a lower bound on the minimum free Euclidean distance which applies to any trellis code.

A more general and tighter version of the bound can be derived, but involves a more detailed proof [15]. In general, this bound shows that random coding bounds can be improved significantly by restricting the set of codes. It also shows that codes with a large increase in the metric at the ends of error events perform well on the average. Finally, it can be used to compare TCM schemes with small constraint lengths without having to search for the corresponding codes.

REFERENCES

[1] G. Ungerboeck, "Channel Coding with Multilevel Phase Signals", IEEE Trans. Inform. Theory, vol. IT-28, pp. 55-67, Jan. 1982.

[2] L.F. Wei, "Trellis-Coded Modulation with Multi-dimensional Constellations", IEEE Trans. Inform. Theory, Vol. IT-33, pp. 483-501, July 1987.

[3] G.D. Forney, Jr., "Coset Codes I: Geometry and Classification", IEEE Trans. Inform. Theory, to appear.

[4] M.F. Rouanne and D.J. Costello, Jr., "A Lower Bound on the Minimum Euclidean Distance of Trellis Coded Modulation Schemes", IEEE Trans. Inform. Theory, to appear.

[5] H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," Ann. Math. Stat. , vol. 23, pp. 493-507, 1952.

[6] R. G. Gallager, "A simple Derivation of the Coding Theorem and Some Applications," IEEE Trans. Inform. Theory , vol. IT-11, pp. 3-18, Jan. 1965.

[7] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory , vol. IT-13, pp. 260-269, Apr. 1967.

[8] G. D. Forney, Jr. "Convolutional Codes II: Maximum Likelihood Decoding", Information and Control , vol. 25, pp. 222-266, July 1974.

[9] D. J. Costello, Jr. "Free Distance Bounds for Convolutional Codes", IEEE Trans. Inform. Theory, vol. IT-20, pp. 356-365, May 1974.

[10] K. J. Larsen, "Short convolutional codes with maximal free distance for rates 1/2, 1/3, and 1/4," IEEE Trans. Inform. Theory, vol. IT-19, pp. 371-372, May 1973.

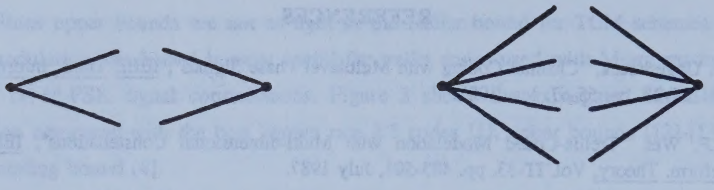
[11] J.A. Heller, "Short constraint length convolutional codes," Jet Propul. Lab., Calif. Inst. Tech., Pasadena, CA. Space Programs Summary 37-54, vol. 3, pp. 171-174, Dec 1968.

[12] G.J. Pottie and D.P. Taylor, "Sphere-Packing Upper Bounds on the Free Distance of Trellis Codes", IEEE Trans. Inform. Theory, to appear.

[13] M. Burnashev and E. Biglieri, "Bounds on the Minimum Distance of Trellis Codes", submitted to IEEE Trans. Inform. Theory, 1986.

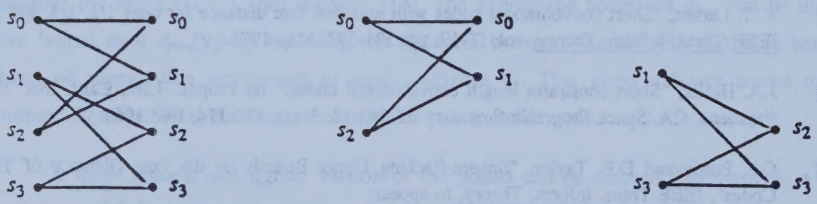
[14] M.F. Rouanne and D.J. Costello, Jr., "Nested Trellis Codes", in preparation.

[15] M.F. Rouanne and D.J. Costello, Jr., "An Expurgated Lower Bound on the Free Distance of Trellis Coded Modulation Schemes", in preparation.



(a) 1 input bit (b) 2 input bits

Figure 1: Adjacent distances for 1 or 2 input bits.


s_0
s_0
s_1
s_1
s_2
s_2
s_3
s_3
s_0
s_0
s_1
s_2
s_1
s_2
s_3
s_3

4-STATE TRELLIS (a) 2 2-STATE SUBTRELLISES (b)

Figure 2: 2-state subtrellises in a 4-state trellis.

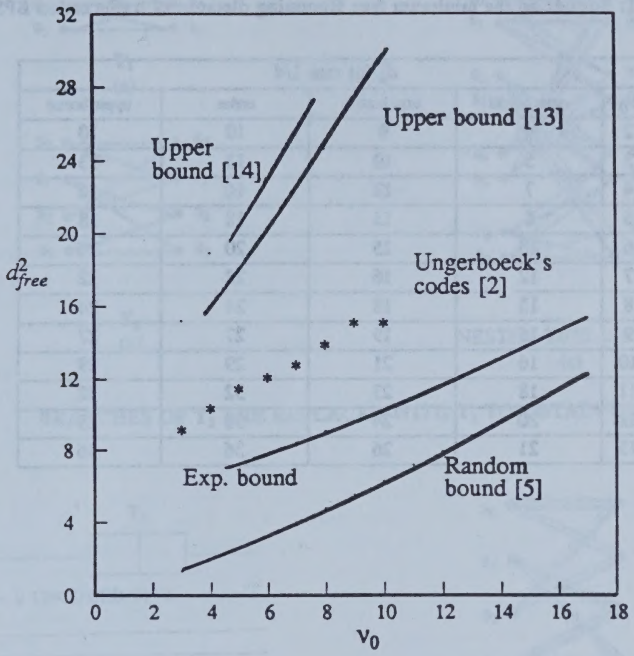


Figure 3: Bounds and on the free Euclidean distance of trellis coded 8-PSK.

Table 1: Bounds on the minimum free Hamming distance of trellis coded BPSK

d_h for rate 1/4				
V_0	rand. bound	exp. bound	codes	upper bound
2	4	9	10	10
3	5	10	13	13
4	7	12	16	16
5	8	13	18	18
6	10	15	20	20
7	12	16	22	22
8	13	18	24	24
9	15	19	27	27
10	16	21	29	29
11	18	23	32	32
12	20	24	33	33
13	21	26	36	36

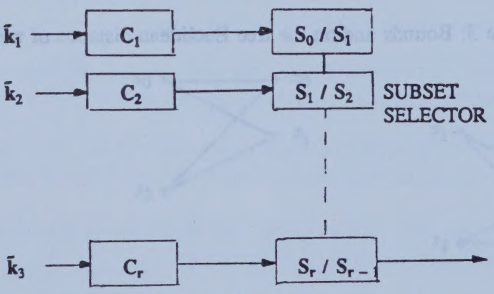
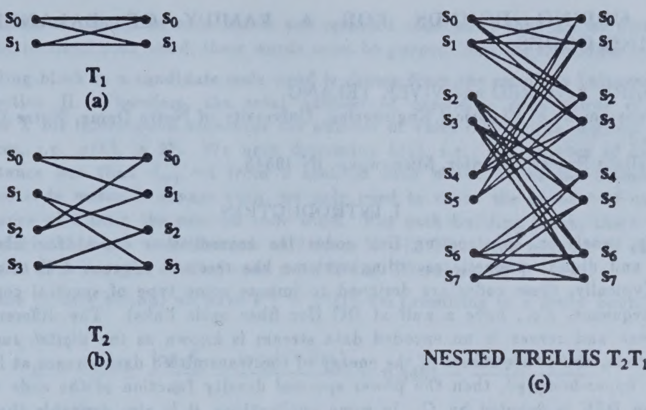
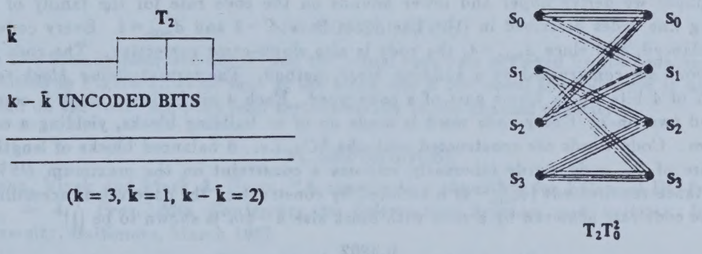

$\bar{k}_1$
C_1
S_0 / S_1
$\bar{k}_2$
C_2
S_1 / S_2
SUBSET
SELECTOR
$\bar{k}_3$
C_r
S_r / S_{r-1}

Figure 4: A generalized Trellis Coded Modulation scheme (nested trellis).


S0
S1
S0
S1
T1
(a)
S0
S1
S2
S3
S0
S1
S2
S3
T2
(b)
S0
S1
S2
S3
S4
S5
S6
S7
S0
S1
S2
S3
S4
S5
S6
S7
NESTED TRELLIS T2T1.
(c)

BRANCHES OF T_2 ARE REPLACED WITH T_1 TO OBTAIN T_2T_1.


k̄
T2
k − k̄ UNCODED BITS
(k = 3, k̄ = 1, k − k̄ = 2)
S0
S1
S2
S3
S0
S1
S2
S3
T2T0²

EXAMPLE OF UNGERBOECK'S TRELLIS CODE.

(d)

Figure 5: Examples of nested trellises.

RANDOM CODING BOUNDS FOR A FAMILY OF BALANCED ERROR-CORRECTING CODES

MARK A. HERRO and VIVEK TELANG
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, and
Tellabs Research Center, Mishawaka, IN 46545.

I. INTRODUCTION

Recently, work on constructing line codes (or transmission codes) for fiber optic digital transmission and digital magnetic recording systems has received a great deal of attention in the literature. Typically, these codes are designed to impose some type of spectral constraints on the transmitted sequence, *e.g.*, have a null at DC (for fiber optic links). The difference between the number of *ones* and *zeroes* in an encoded data stream is known as the *digital sum variation*, or *DSV*, of the code, and is a measure of the energy of the transmitted data stream at DC. If the *DSV* for a code is upper-bounded, then the power spectral density function of the code vanishes at DC. The maximum *DSV* is denoted by C. In some applications, it is also desirable that the line codes provide some error-correcting or error-detecting capability as well.

II. DESCRIPTION OF CODE

In this paper we derive upper and lower bounds on the code rate for the family of balanced error-correcting line codes described in [1]. The codes have $C=2$ and $d_{\min}=4$. Every code word in the code is balanced, and since $d_{\min}=4$, the code is also single-error correcting. The code words in the proposed code are constructed by a *building block* method. The term *building block* refers to a balanced block of 4 bits which forms part of a code word. Each 4 bit building block is composed of two *zeroes* and two *ones*. Every code word is made up of m building blocks, yielding a code word length $n=4\times m$. Code words are constructed with the ${}^4C_2$, *i.e.*, 6 balanced blocks of length 4. The balanced nature of the code words inherently imposes a constraint on the maximum *DSV* for the code. The distance requirement ($d_{\min}=4$) is satisfied by constructing the code words according to certain rules. The code rate achieved by a code with block size $n=4m$ is shown to be [1]

$$R = 0.6462 - \frac{0.3962}{m} \text{ bits/symbol}, \tag{1}$$

resulting in a code rate that tends to 0.6462 bits/symbol for large values of $n=4m$.

III. CHANNEL CAPACITY (UPPER BOUND ON CODE RATE)

The method of construction used in designing the family of codes leads to inherent restrictions on the allowable sequences of transmitted symbols. For the code under consideration the *DSV* is zero after every block of 4 bits. The resultant input restricted channel can be represented by the finite state model depicted in Figure 1. The state variables are the position in the 4-bit block, *viz.* a number from 1 through 4, and the value of the *DSV* after that bit position.

We define $\mathbf{W}$ to be a square matrix, (called the connectivity matrix), such that the entry $\mathbf{W}_{ij}$ equals 1 if a transition is allowed from state i to state j and 0 otherwise. For the finite state model shown in Figure 1, $\mathbf{W}$ is a 8×8 matrix. Shannon [2] has shown that the channel capacity of an input restricted channel whose connectivity matrix is $\mathbf{W}$, is given by $H = \log_2\lambda$, where λ is the largest real eigenvalue of the matrix $\mathbf{W}$, *i.e.*, the largest real root of the equation $|\mathbf{W}-\lambda\mathbf{I}| = 0$. The largest real eigenvalue of $\mathbf{W}$ defined above is 1.5651. Thus, the capacity of this particular input restricted channel is

$$H = \log_2(1.5651) = 0.6462 \text{ bits/symbol}. \tag{2}$$

V. GILBERT TYPE LOWER BOUND ON CODE RATE

We now derive a Gilbert type lower bound on the code rate for these codes. The general form of these bounds is given by $f(n) \geq g(k)(1 + \sum_{i<d_{\min}} h(i))$, where

$f(n) =$ the total number of candidate code words of length n,
$g(k) =$ the number of retained code words, such that the resulting set of code words has minimum distance $d_{\min}$, and

$h(i) =$ the number of candidate code words per retained code word that are at distance $i < d_{\min}$ from the retained code word; these words must be purged from the candidate set.

Each building block in a candidate code word is chosen from the set of six balanced 4-bit blocks described in Section II. Therefore, the total number of candidate code words of length n is $f(n) = 6^m$. For k bit information sequences the number of valid code words required to encode all possible sequences, *i.e.* $g(k)$, is 2^k. We next determine $h(i)$, *i.e.*, the number of candidate code words at a distance less than $d_{\min}=4$ from a selected code word. Since the Hamming distance between balanced code words is always even, we only need to count the number of candidate code words at a distance of 2 from the selected code word. For each building block, there are exactly 4 other building blocks that are at a distance of 2 from it [1]. Thus, the number of candidate code words at a distance of 2 from the selected code word is $4m$. Therefore, $h(i) = 4m$.

Applying the Gilbert bound, we have $6^m \geq 2^k(1+4m)$ resulting in a lower bound on the code rate given by

$$R = \frac{k}{4m} \leq \frac{\log_2 6}{4} - \frac{\log_2(1+4m)}{4m} = 0.6462 - \frac{\log_2(1+4m)}{4m} \tag{3}$$

Note that as the block size n, (and therefore m,) goes to infinity, the lower bound on the code rate tends to 0.6462, which from (2), is also the capacity of the channel. This phenomenon can be explained by the fact that with increasing block size, $n = 4m$, the number of candidate code words increases faster than the number of code words that must be expurgated for each selected code word.

IV. CONCLUSIONS

The two bounds described above, and the actual code rate achieved by the code under discussion [1], are plotted in Figure 2. Note that as the block size n increases, the code rate R and the Gilbert lower bound on the code rate both tend asymptotically to the channel capacity.

VI. REFERENCES

[1] Telang, Vivek and Mark A. Herro, "A constructive algorithm for balanced DC-free codes with $d_{\min} = 4$," *Proc. 20th Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore, March 1987.

[2] Shannon, C. E., "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, July 1948.

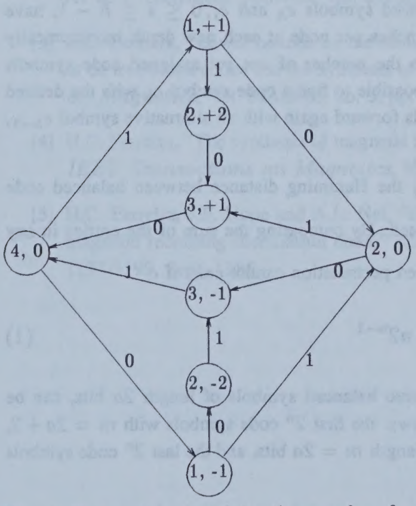


Fig. 1. Finite state model for the encoder of a balanced code with $DSV = 0$ after every 4-bit block.

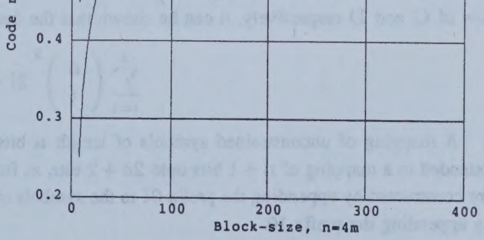


Fig. 2. Bounds on code rate R vs. block size n

ON GENERALIZED ERROR CORRECTING TRELLIS CODES WITH BALANCED BINARY SYMBOLS

1) H.C. FERREIRA
3) A.L. NEL
Rand Afrikaans University
Johannesburg, South Africa

2) D.A. WRIGHT
San Diego State University
San Diego, California

Balanced or dc free binary block codes (i.e. codes with equal numbers of ones and zeros in each codeword) find widespread application on repeatered PCM channels, optical communication channels and digital magnetic or optical recorders. Recently the synthesis of balanced binary block codes with error correction capabilities, was presented in [1] to [3]. An error correcting trellis code with balanced binary symbols was described in [4] and a similar code with runlength constrained symbols in [5]. These two trellis codes were synthesized by mapping constrained binary symbols onto the state systems of linear convolutional codes serving as base codes. However, every time a new base code was chosen, a new mapping, which can become an elaborate procedure for base codes with large free distances, had to be devised.

Here we investigate the invariable mapping of the unconstrained n bit code symbols of any $R = k/n$ linear convolutional code onto 2^n balanced code symbols of length $m = 2a$ bits. This mapping has the property that the Hamming distance between any two balanced symbols is greater or equal than the Hamming distance between the unconstrained binary symbols mapped onto them - consequently, the free distance of the resulting $R' = k/m$ balanced trellis code can be no less than the free distance of the base code, irrespective of the convolutional code chosen as base code. We note that there are $N(m) = \binom{2a}{a}$ elements in the set of balanced code symbols. We only investigate mappings where $n = \lfloor \log_2 N(m) \rfloor$.

A permutation of balanced symbols with the abovementioned distance property, can be found by means of a tree search, similar to the Fano algorithm. Firstly, a square symmetrical matrix $\mathbf{D}$ with elements d_{ij} the Hamming distances between unconstrained binary code symbols i and j, $0 \le i, j \le 2^n - 1$, is computed. The tree search proceeds from depth 0 to depth $2^n - 1$ and attempts at depth h to find a balanced code symbol c_h such that the Hamming distances e_{hi}, between the balanced symbols c_h and $c_i, 0 \le i \le h-1$, have $e_{hi} \ge d_{hi}$. The tree has the property that the number of branches per node at each new depth monotonically decreases by one, since the number of branches is equal to the number of not yet assigned code symbols remaining in the set of balanced symbols. Whenever it is impossible to find a code symbol c_h with the desired property, the tree search backs up to depth $h-1$, and proceeds forward again with an alternative symbol c_{h-1}, and so forth, similar to the Fano algorithm.

Before the tree search, the matrix $\mathbf{G}$ with elements g_{ij} the Hamming distance between balanced code symbols c_i and c_j, $0 \le i, j \le \binom{2a}{a} - 1$, can be computed. By comparing the sum of the entries in any row of $\mathbf{G}$ and $\mathbf{D}$ respectively, it can be shown that the desired permutation cannot exist if

$$\sum_{i=1}^{a} \binom{a}{i}^2 2i < n2^{n-1} \tag{1}$$

A mapping of unconstrained symbols of length n bits onto balanced symbols of length $2a$ bits, can be extended to a mapping of $n+1$ bits onto $2a+2$ bits, as follows: the first 2^n code symbols with $m = 2a+2$, are constructed by appending the prefix 01 to the symbols of length $m = 2a$ bits, and the last 2^n code symbols by appending the prefix 10.

Various tree searches were conducted with the objective to minimize the maximum runlength, l, when any two code symbols are concatenated. The best results are summarized in the following table. The invariable mappings are specified by mapping the vector of unconstrained n bit code symbols, $\bar{P} = (0, 1, \ldots, 2^n - 1)$ onto the vector $\bar{Q}$ containing the m bit balanced code symbols.

Constrained Trellis Codes: Balanced Symbols
(Hexadecimal notation for $\bar{Q}$)

m	$N(m)$	n	Highest rates		Synthesis	Permutation $\bar{Q}$	(b, l, C)
			Base code, $R = k/n$	Constrained code, $R' = k/m$			
4	6	2	1/2	1/4	Tree Search	(5, 9, A, 6)	(0, 1, 1)
6	20	4	3/4	3/6	Tree search	(B, D, E, 19, 16, 13, 1A, 15, 26, 29, 2A, 2C, 32, 25, 34, 31)	(0, 3, 2)

b = minimum runlength in NRZI = (minimum runlength in NRZ) - 1
l = maximum runlength in NRZI = (maximum runlength in NRZ) - 1
C = upperbound on accumulated charge

References

[1] M.A. Herro and L. Hu, "Error-correcting line codes," presented at the twenty-third Allerton Conference on Communication Control and Computing, October 1985, Monticello, Illinois.

[2] E.E. Bergmann, A.M. Odlyzko and S.H. Sangani, "Half weight block codes for optical communications," AT & T *Technical Journal*, Vol. 65, pp. 85-93, May/June 1986.

[3] H.C. Ferreira, "Lower bounds on the minimum Hamming distance achievable with runlength constrained or dc free block codes and the synthesis of a (16, 8) $d_{min} = 4$ dc free block code," *IEEE Transactions on Magnetics*, Vol. MAG-20, no. 5, pp. 881-883, Sept. 1984.

[4] H.C. Ferreira, "The synthesis of magnetic recording trellis codes with good Hamming distance properties," *IEEE Transactions on Magnetics*, Vol. MAG-21, no. 5, pp. 1356-1358, Sept. 1985.

[5] H.C. Ferreira, J.F. Hope and A.L. Nel, "Binary rate four eighths, runlength constrained, error correcting magnetic recording modulation code," *IEEE Transactions on Magnetics*, Vol. MAG-22, no. 5, pp. 1197-1199, Sept. 1986.

EFFICIENT MAXIMUM-LIKELIHHOD DECODING OF THE EXTENDED NORDSTROM-ROBINSON CODE

AYYOOB D. ABBASZADEH & C.K. RUSHFORTH
Department of Electrical Engineering
University of Utah, SLC, UT 84102

ABSTRACT

A fast maximum-likelihood (ML) decoding algorithm for the extended Nordstrom-Robinson (16,256,6) nonlinear code in the additive white Gaussian noise channel is developed using an approach recently described by Conway and Sloane [1]. The desired partitioning of the Nordstrom-Robinson code into cosets is achieved using a subcode isomorphic to the (4,3) even-parity code. The resulting algorithm requires about one-eighth as many operations as direct ML decoding and is readily adaptable to pipelining and parallelism[†].

I. INTRODUCTION

Maximum likelihood (ML) decoding of a binary block code **L** of length **n** in the presence of additive white Gaussian noise maps an arbitrary point **x** in $\mathbf{R}^n$ onto that codeword **u** closest to **x** in Euclidean distance. The most direct approach to ML decoding is simply to compute the distance from the received vector **x** to each codeword and to select the closest. The complexity of this procedure grows rapidly with the size of the code, and it soon becomes impractical.

Conway and Sloane [1] describe the following exact ML decoding algorithm that can be much more efficient than the direct procedure if the code in question has a subcode of small index that can itself be rapidly decoded. Suppose that the code **L** contains a subcode **B**. Let $\rho^{(j)}$, j=0,1,.., t-1, be the jth coset leader of **B** in **L**. With **L** defined over {1,-1}, we can write

$$\mathbf{L} = \bigcup_{j=0}^{t-1} \left(\rho^{(j)} * \mathbf{B} \right) \qquad (1)$$

where * denotes component-by-component multiplication. Using this structure, we can determine the codeword with the largest inner product as follows [1]:

1. Compute $r^{(j)} = \rho^{(j)} * \mathbf{x}$ for j = 0, 1, . . ., t-1.
2. Use the decoder for **B** to find $y^{(j)}$, the vector in **B** closest to $r^{(j)}$. Store $y^{(j)}$ and the associated inner product $ip^{(j)}$.
3. Select the largest of the inner products computed in step 2, say $ip^{(m)}$.
4. The codeword in **L** closest to **x** is $\rho^{(m)} * y^{(m)}$.

We show how this decoding approach can be applied to the ML decoding of the extended Nordstrom-Robinson (NR) code.

II. STRUCTURE OF THE EXTENDED NORDSTROM-ROBINSON CODE

The extended NR (16,256,6) code can be constructed as a subset of the permuted codewords of the extended Golay (24,12) code with the first 8 coordinates deleted [2]. Based on this construction, the extended NR code can be expressed in terms of the first-order Reed-Muller RM(1,4) code of length 16 and its corresponding eight cosets, i.e.

$$NR = \bigcup_{j=0}^{7} \left(\rho_{RM}^{(j)} * RM(1,4) \right) \qquad (2)$$

[†]This work was supported by the Unisys Corporation under University of Utah contract number 5-20555.

where

$$\{\rho_{RM}\} = \begin{bmatrix} 0000\ 0000\ 0000\ 0000 \\ 0100\ 0010\ 0010\ 1011 \\ 0000\ 0101\ 0011\ 1001 \\ 1000\ 0100\ 1000\ 1011 \\ 0100\ 0001\ 0111\ 0010 \\ 0000\ 0110\ 0101\ 0011 \\ 0000\ 0011\ 1001\ 1010 \\ 0100\ 0100\ 1110\ 0001 \end{bmatrix} \qquad (3)$$

III. DECODING OF RM(1,m) CODES

In (+1,-1) notation the codewords of RM(1,m) are the rows of the matrix

$$RM(1,m) = \begin{bmatrix} H_N \\ \hline -H_N \end{bmatrix} \qquad (4)$$

where H_N is a normalized Hadamard matrix of order $N=2^m$ that can be decomposed into a product of logN identical matrices Q_N. Using this construction and the fact that multiplication of the received vector **x** by Q_N requires only N real additions, fast Hadamard decoding of RM(1,m) code would require NlogN additions and N-1 comparisons.

The codewords of RM(1,m) can also be constructed from the codewords of RM(1,m-1) as follows:

$$RM(1,m)=\{\ (\mu,\mu\ \oplus\upsilon)\ \big|\ \mu\in RM(1,m-1)\ \text{and}\ \upsilon\in RM(0,m-1)\ \} \qquad (5)$$

Based on this construction, we can identify a subcode of RM(1,m) which is isomorphic to the even-parity (4,3) code . The coset leaders of this subcode are 2^{m-2} words of the form (μ,μ,μ,μ), with μ a row of H_2m-2. The decoding algorithm for this subcode is as follows:

1. Receive $\mathbf{x}=(x_N,\ldots,x_1)$ and calculate four S_k's as: $S_k=\sum_{i=0}^{\alpha-1} x_{\alpha*k-i}$, where $\alpha=N/4$.

2. If an even number of S_k's are positive, decode **x** as $U_1^{\alpha}U_2^{\alpha}U_3^{\alpha}U_4^{\alpha}$ and set $ip=\sum_{k=1}^{4}|S_k|$.

Otherwise set $|S_j|=\min_k\{|S_k|\}$, $U_j=-\mathrm{sgn}(S_j)$, decode **x** as above, and set $ip=\sum_{k=1}^{4}|S_k|-2*S_j$.

IV. DECODING THE NORDSTROM-ROBINSON CODE

To decode the extended NR code, we can apply fast Hadamard decoding to each of the eight modified input vectors $\mathbf{x}*\rho^{(j)}_{RM}$, selecting the codeword with the maximum inner product. This approach will require 8(NlogN)=512 additions and 8(15)=120 comparisons. This compares favorably with the 1920 additions and 127 comparisons required by direct ML decoding of the NR code.

A further reduction in the number of computations required can be achieved by taking **B** to be the subcode of order eight of RM(1,4) described previously. Decoding using this partition is facilitated by precomputing and storing the absoulte values of the S_k's (k=1,..,4) and then using these precomputed values to perform the decoding of the subcode 32 times. This approach requires 72 (precomputation) + 128 (decoding) = 200 additions and 8(4*3)=96 comparisons. Hence, this algorithm requires only about half as many computations as the algorithm based on conventional Fast Hadamard decoding, and about one-eighth as many as direct ML decoding. Moreover, it is suited for pipelining and parallel implementation.

REFERENCES

[1]J.H. Conway and N.J.A. Sloane, " Soft Decoding Techniques for Codes and Lattices, Including the Golay Code and the Leech Lattice," *IEEE Trans. Inform. Theory* , Vol. IT-28, No. 1, pp. 41-50, Jan. 1986.

[2]F.J. MacWillams and N.J.A. Sloane, **The Theory of Error Correcting Codes**, North-Holland Publishing Company, Amsterdam, 1977.

SOFT DECISION DECODING OF CONCATENATED CODES

AMER A. HASSAN
WAYNE E. STARK
The Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI 48109.

ABSTRACT

The correcting properties of concatenated codes with parallel decoding is investigated. We are concerned with the situation where no quantization is performed at the output of the demodulator; i.e., the additive channel case. An assesment of the correcting properties is obtained by the analysis of a game theoretic problem with the channel and decoder as players. An optimal decoder strategy is found for a fixed number of decoder branches.

1. INTRODUCTION

In this paper we consider a parallel decoding algorithm for concatenated codes (see Fig. 1.). Unlike the paper by Zyablov [2] we consider a soft decision inner decoders which processes the received vector and either outputs an estimate or an erasure. Each of the inner decoders has a different threshold for deciding when to erase. Basically if the received vector is within (Euclidean) distance Δ_i to some codeword the i-th decoder will output the information symbols of that codeword; otherwise the decoder will erase the codeword and pass the erasure to the outer decoder.

Let z be the number of inner and outer decoders. Then for each z we determine the optimal choice of $\Delta_i, i = 1, 2, ..., z$, to maximize the minimum Euclidean distance correctable by the code. Furthermore, we show that for reasonable value of z (3 or 4) this decoding algorithm can correct close to the minimum distance of the concatenated code.

The correcting capability is evaluated by analyzing a game situation with the channel and decoder as opponents. The strategy of the channel consists of all noise vectors. The strategy for the decoder consists of the thresholds for declaring erasures in each decoding branch. The objective of the channel is to (for reasons which will be obvious later) minimize the Euclidean length of the noise it adds to the transmitted signal necessary to cause a decoding error. The objective of the decoder is to choose the appropriate thresholds to maximize the noise length necessary to cause an error.

The paper is structured in the following way. In Section 2, we briefly discuss the basic concepts of concatenated coding and some properties of its error correcting capability. In Section 3 we present a game theoretic formulation of the problem and investigate the channel strategy in detail. Finally, the optimal decoder strategy is found in Section 4 for a fixed number of decoder branches.

2. CONCATENATED CODING

Concatenated codes can be represented as the direct product of codes of lengths n_1 and n_2 with code minimum Hamming distances d_{1H} and d_{2H} respectively. These codes are refered to as inner-codes and outer-codes. Formally, let the outer encoding be characterized by a mapping $f : M \longrightarrow X^{n_2}$, where M is a message set and X is the alphabet of the outer code; also let the inner-encoding be $g : X \longrightarrow U^{n_1}$, then concatenation is defined by $F \triangleq g^{n_2} f : M \longrightarrow U^{n_1 n_2}$.

For the M-ary input-output channel, designing a decoder which employs the maximum possible error-correcting capability [1] (i.e., one that corrects all error patterns of weight less than $\frac{d_{1H}d_{2H}}{2}$) is a nontrivial task. For instance, if the inner code is used to correct $\lfloor\frac{d_{2H}-1}{2}\rfloor$ errors and the outer code to correct $\lfloor\frac{d_{1H}-1}{2}\rfloor$ errors, then the decoder $f^{-1}(g^{-1})^{n_2}(u^{n_1 \times n_2})$ can correct only about $\frac{d_{1H}d_{2H}}{4}$ errors. For this case Zyablov [2] found an algorithm which has the maximum possible error-correcting capability, at the expence of increasing the complexity of the decoder, and Ericson [3] has a simplified treatment of the algorithm. A more general treatment of concatenated codes for the M-ary input-output channel can be found in [4].

3. GAME THEORETIC FORMULATION AND CHANNEL STRATEGY

Consider the parallel decoding algorithm such that the inner decoder of the i-th branch decodes all vectors within Euclidean distance Δ_i of some codeword, $i = 1, 2, ..., z$, and detects errors otherwise, $\Delta_i \in (0, \frac{d_{1E}}{2})$, where z is the number of decoder branches. Further let d_{1E} = minimum Euclidean distance of the inner code.

The foregoing analysis is restricted for outer codes defined over GF(q), for all $q = 2^m$, $m \geq 1$, and binary inner codes. Then d_{1E} and d_{1H} are related as follows

$$d_{1E} \quad = \begin{cases} 2\sqrt{d_{1H}} & \text{for antipodal signalling;} \\ \sqrt{2d_{1H}} & \text{for orthogonal signalling.} \end{cases} \tag{1}$$

The outer code corrects any set of e errors and τ erasures if $2e + \tau < d_{2H}$. Of the z possible decoding results, the one that has the smallest Euclidean distance from the received combination is taken as the final result.

Now let the transmitted matrix be given by $\underline{s} = (s_{ij}, 1 \leq i \leq n_2, 1 \leq j \leq n_1)$ (for binary codes, $s_{ij} \in \{-1, +1\}$) then the received vector is $\underline{y} = \underline{s} + \underline{\nu}$ where $\underline{\nu} = (\nu_{ij}, 1 \leq i \leq n_2, 1 \leq j \leq n_1)$ is the additive noise matrix; that is $y_{ij} = s_{ij} + \nu_{ij}$, where ν_{ij} is noise. The square of the Euclidean distance between $\underline{s}$ and $\underline{y}$ is

$$t \triangleq d_E^2(\underline{s}, \underline{y}) = \sum_{i=1}^{n_2} \sum_{j=1}^{n_1} \nu_{ij}^2 \tag{2}$$

If d_E and d_H are, respectively, the Euclidean distance and Hamming distance of the concatenated code, then for any transmitted product codes $\underline{s}$ and $\underline{s}'$ we have

$$d_E^2 = \sum_i \sum_j (s_{ij} - s'_{ij})^2 \geq \begin{cases} 4d_{1H}d_{2H} & \text{antipodal signalling;} \\ 2d_{1H}d_{2H} & \text{orthogonal signalling} \end{cases} \tag{3}$$

or equivalently as

$$d_E \geq \begin{cases} 2\sqrt{d_{1H}}.\sqrt{d_{2H}} & = d_{1E}.\sqrt{d_{2H}}; \\ \sqrt{2d_{1H}}\sqrt{d_{2H}} & = d_{1E}\sqrt{d_{2H}}. \end{cases} \tag{4}$$

Hence, we are interested in finding the bounded-distance decoder which decodes correctly any vector within Euclidean distance $\frac{d_{1E}\sqrt{d_{2H}}}{2}$ from the transmitted one.

To find the best decoding method, we treat the problem as a game with two players. Player 1, called the channel, strategies consist of all noise vectors $\underline{\nu}$. Player 2, called the decoder, strategies consist of all vectors Δ_i, $0 \leq \Delta_1 < ... < \Delta_z < \frac{d_{1E}}{2}$. Player 1 chooses $\underline{\nu}$ to minimize the noise

---

[1] Note that the minimum Hamming distance of an $n_2 \times n_1$ code $\geq d_{1H}d_{2H}$, with equality if and only if each of the minimum-weight (all codes are linear) outer code symbols is encoded by the inner encoder to a minimum-weight inner codeword.

length t to gurantee each decoding branch is not decoded correctly. Player 2 chooses the set $\Lambda_z = \{\Delta_1, \Delta_2, ..., \Delta_z\}$ to maximize the minimum noise vector length that will cause an error. By convention we let $\Delta_0 = 0$ and $\Delta_{z+1} = d_1 - \Delta_z$. Hence, assotiated with the game are two programs:
Program I (Decoder's Program)

$$\gamma = \max_{\Lambda_z} \min_{\underline{\nu}} t$$

where the minimum over $\underline{\nu}$ is such that the errors in the received vector cannot be corrected by the decoder (i.e., the output of each decoder branch is incorrect).
Program II (Channel's Program)

$$\gamma\prime = \min_{\underline{\nu}} \max_{\Lambda_z} t$$

where the minimum over $\underline{\nu}$ is also conditioned on the decoder output being incorrect. We will call t the channel loss.

It is clear from the above programs that $\gamma \leq \gamma\prime$; therefore, for a pessimistic analysis (observed by the decoder) we are interested in investigating program I which we do in several steps; however, we need some additional definitions that are convenient in finding optimal solutions to our program. Define

$$x_m = \text{ number of rows such that} \sum_{j=1}^{n_1} \nu_{ij}^2 = m.$$

Let $N(m)$ be a counting measure with $N(m) = 1$ if $x_m > 0$ and $N(m) = 0$, otherwise; then

$$t = \sum_{m \in [0,\infty)} m x_m \equiv \int_0^\infty m x_m dN(m). \tag{5}$$

The above summation is well defined because $x_m = 0$ except for a finite number of m's.

At this point we note that the channel should always choose the vector $\underline{\nu_i} = (\nu_{i1}, \nu_{i2}, ..., \nu_{in_1})$ in the direction of the minimum weight (inner) codeword. In this case the channel strategy then is to choose how many of the vectors there are of a given length, that is $\underline{x} = \{x_m : m \in [0, \infty)\}$. This also specifies the number of erasures τ_k and errors e_k in the outer code for the k-th decoding branch. We introduce the notation $\tau = (\tau_1, \tau_2, ..., \tau_z)$ and $e = (e_1, ..., e_z)$.

The values of τ_k and e_k are related to the coordinates of $\underline{x}$ by the expressions

$$\tau_k + e_k = \int_{\Delta_k^2}^\infty x_m dN(m) \quad k = 1, 2, ..., z. \tag{6}$$

For a Δ_k distance correcting code, error detection will certainly occur when $\Delta_k^2 \leq \sum_{j=1}^{n_1} \nu_{ij}^2 \leq (d_{1E} - \Delta_k)^2$. Either error detection or incorrect decoding will occur if $\sum_{j=1}^{n_1} \nu_{ij}^2 > (d_{1E} - \Delta_k)^2$. Therefore we obtain the following bounds

$$\tau_k \geq \int_{\Delta_k^2}^{(d_{1E}-\Delta_k)^2} x_m dN(m) \text{ and } e_k \leq \int_{(d_{1E}-\Delta_k^+)^2}^\infty x_m dN(m) \quad k = 1, ..., z. \tag{7}$$

From (5) we can write the channel loss as

$$\begin{aligned} t &= \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) + \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} m x_m dN(m) \\ &\geq \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + (d_{1E} - \Delta_1^+)^2 \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m) \\ &\geq \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + (d_{1E} - \Delta_1^+)^2 e_1 \end{aligned} \tag{8}$$

with equality if $e_1 = \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m)$; this occurs when $x_m = e_1$ for $m = (d_{1E} - \Delta_1^+)^2$ and $x_m = 0$ for $m > (d_{1E} - \Delta_1^+)^2$. Similarly, going backward in equation (8) we conclude that the channel loss is minimum when equality is attained in (7) for $k = 1, 2, ..., z$. Then in the case of optimal channel strategy, we can rewrite (7) in a form more convenient for later use, as follows:

$$\begin{aligned} e_k &= \int_{(d_{1E}-\Delta_k^+)^2}^{(d_{1E}-\Delta_{k-1})^2} x_m dN(m) + \int_{(d_{1E}-\Delta_{k-1}^+)^2}^{\infty} x_m dN(m) \\ &= \int_{(d_{1E}-\Delta_k^+)^2}^{(d_{1E}-\Delta_{k-1})^2} x_m dN(m) + e_{k-1} \quad k = 2, 3, ..., z \end{aligned} \tag{9}$$

$$\begin{aligned} \tau_k &= \int_{(\Delta_k)^2}^{(\Delta_{k+1}^-)^2} x_m dN(m) + \int_{\Delta_{k+1}^2}^{(d_{1E}-\Delta_{k+1})^2} x_m dN(m) + \int_{(d_{1E}-\Delta_{k+1}^+)^2}^{(d_{1E}-\Delta_k)^2} x_m dN(m) \\ &= \int_{\Delta_k^2}^{(\Delta_{k+1}^-)^2} x_m dN(m) + \tau_{k+1} + e_{k+1} - e_k \quad k = 1, 2, ..., z-1 \end{aligned} \tag{10}$$

Also,

$$e_1 = \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m), \text{ and } \tau_z = \int_{\Delta_z^2}^{(d_{1E}-\Delta_z)^2} x_m dN(m) \tag{11}$$

It follows from (9) and (10) that

$$e_k \leq e_{k+1}, \text{and } \tau_k \geq \tau_{k+1} \quad k = 1, 2, ..., z-1.$$

The above inequalities will be used as optimization constraints later in this section. We estimate the channel loss $\gamma(\Lambda_z, \tau, e)$ in the case of its optimal strategy under the condition that there are τ_k erasures and e_k errors in the outer code in the k-th decoding branch; that is,

$$\gamma(\Lambda_z, \tau, e) = \min_{\underline{x}} \int_0^{\infty} m x_m dN(m) \tag{12}$$

where the minimum is over $\underline{x}$ satisfying (9)-(11). Before we proceed, we note that

$$\min \int_{j_1}^{j_2} m x_m dN(m) = j_1 a \tag{13}$$

where the minimum is over all x_m , $j_1 \leq m \leq j_2$ such that $\int_{j_1}^{j_2} x_m dN(m) = a$. Furthermore, this occurs when $x_{j_1} = a$ and $x_j = 0$ for $j_1 < j \leq j_2$. Then (12) can be specified as follows using (9)-(11) and (13):

$$
\begin{aligned}
\gamma(\Lambda_z,\boldsymbol{\tau},\boldsymbol{e}) &= \min_{\boldsymbol{x}} \int_0^{\infty} m x_m dN(m) \\
&= \min_{\boldsymbol{x}}\{\int_0^{(\Delta_1^-)^2} m x_m dN(m) + \int_{\Delta_1^2}^{(\Delta_2^-)^2} m x_m dN(m) \\
&\quad + \int_{\Delta_2^2}^{(\Delta_3^-)^2} m x_m dN(m) + \ldots + \int_{\Delta_{z-1}^2}^{(\Delta_z^-)^2} m x_m dN(m) \\
&\quad + \int_{\Delta_z^2}^{(\Delta_{z+1}^-)^2} m x_m dN(m) + \int_{(d_{1E}-\Delta_z^+)^2}^{(d_{1E}-\Delta_{z-1})^2} m x_m dN(m) + \ldots \\
&\quad + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) + \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} m x_m dN(m)\} \\
&= \sum_{k=1}^{z}\{(\Delta_k^2-\Delta_{k-1}^2)\tau_k + (2\Delta_k^2-\Delta_{k-1}^2-\Delta_{k+1}^2 \\
&\quad +2d_{1E}\Delta_{k+1}-2d_{1E}\Delta_k)e_k\} \qquad (14) \\
&\equiv \sum_{k=1}^{z}\gamma^{(k)}(\tau_k,e_k)
\end{aligned}
$$

$$(15)$$

From (15) we can determine the minimum channel loss in the case of an optimal strategy under the condition that the $k-$th decoding trial has τ_k erasures and e_k errors. Next we use this result to determine the channel loss $\gamma(\Lambda_z,\boldsymbol{\tau})$ in the case of the optimal channel strategy under the condition that τ_k erasures occur in the $k-th$ decoding trial and decoding is unsuccessful. For all the decoder branches to be unsuccessful, it is necessary and sufficient that the following z inequalities hold:

$$e_k \geq \max(\frac{d_{2H}-\tau_k}{2},0) \quad k=1,2,\ldots,z. \tag{16}$$

$$\text{and } \gamma(\Lambda_z,\boldsymbol{\tau}) = \min_{\boldsymbol{e}}\gamma(\Lambda_z,\boldsymbol{\tau},\boldsymbol{e}) \tag{17}$$

where the minimum is taken over $\boldsymbol{e}$ satisfying (16). From (16) and (17) we have

$$\gamma(\Lambda_z,\boldsymbol{\tau}) = \sum_{k=1}^{z}\min_{e_k}\gamma^{(k)}(\tau_k,e_k) = \sum_{k=1}^{z}\gamma^{(k)}(\tau_k); \tag{18}$$

where

$$\gamma^{(k)}(\tau_k) = \begin{cases} (\Delta_k^2-\Delta_{k-1}^2)\tau_k + b(k)\lfloor\frac{d_{2H}-\tau_k+1}{2}\rfloor & 0\leq\tau_k\leq d_{2H} \\ (\Delta_k^2-\Delta_{k-1}^2)\tau_k & \tau_k\geq d_{2H} \end{cases}, \tag{19}$$

where $b(k)=2\Delta_k^2-\Delta_{k-1}^2-\Delta_{k+1}^2+2d_{1E}\Delta_{k+1}-2d_{1E}\Delta_k$. We use (19) to minimize over all vectors $\boldsymbol{\tau}$ such that $\tau_k \geq \tau_{k+1}$. Then we have

$$\gamma(\Lambda_z) = \min_{\boldsymbol{\tau}}\gamma(\Lambda_z,\boldsymbol{\tau}) = \min_{d_{2H}\geq\tau_1\geq 0}\{\gamma^{(1)}(\tau_1) + \min_{\tau_1\geq\tau_2}\{\gamma^{(2)}(\tau_2) + \ldots + \min_{\tau_{z-1}\geq\tau_z}\gamma^{(z)}(\tau_z)\}\ldots\}\}. \tag{20}$$

The solution to (20) is obtained by the help of the following lemma which we state without a proof.

Lemma: For τ some positive integer, and $0 < a < b$, we have

$$\begin{aligned}\min_{y\le\tau\le x}\{a\tau + b\lfloor(d-\tau+1)/2\rfloor\} &= \frac{db}{2} + \min_{y\le\tau^*\le x}\tau^*(2a-b)/2 \qquad (21)\\ &= db/2 + \frac{1}{2}\min\{x^*(2a-b), y^*(2a-b)\}\end{aligned}$$

where $\tau^* \in \Upsilon$, $x^* = \max\{\tau \in \Upsilon : x \ge \tau\}$, and $y^* = \min\{\tau \in \Upsilon : y \le \tau\}$; Υ being the set of values of τ for which $\tau \equiv d \pmod 2$.

According to the lemma, the minimum in (20) occurs at point $\boldsymbol{\tau}$ whose elements satisfy

$$\tau_1 \equiv \tau_2 \equiv \dots \equiv \tau_z \equiv d_{2H} \pmod 2 \qquad (22)$$

Using (21) and (22) permits us to write (20) as

$$\begin{aligned}\gamma(\Lambda_z) &= \sum_{k=1}^{z}\frac{d_{2H}}{2}b(k) + \min_{\tau^*}\sum_{m=1}^{z}\tau_m^*(\Delta_{k+1}^2 - \Delta_{k-1}^2 + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2\\ &= \frac{d_{2H}(d_{1E}-\Delta_1)^2}{2} + \min_{\tau^*}\sum_{m=1}^{z}\tau_m^*(\Delta_{k+1}^2 - \Delta_{k-1}^2 + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2. \qquad (23)\end{aligned}$$

Two cases for d_{2H} are considered.
CASE I: $d_{2H} \equiv 0 \pmod 2$, then according to lemma the minimum is attained at one of the $z+1$ points $\boldsymbol{\tau} \in S = \{(0,0,\dots,0),(d_{2H},0,\dots,0),(d_{2H},d_{2H},0,\dots,0),\dots,(d_{2H},\dots,d_{2H})\}$ then

$$\begin{aligned}\gamma(\Lambda_z) &= \frac{d_{2H}(d_{1E}-\Delta_1)^2}{2} + \min_{\tau^*}\sum_{m=1}^{z}\tau_m^*(\Delta_{k+1}^2 - \Delta_{k-1}^2 + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2\\ &= \frac{d_{2H}(d_{1E}-\Delta_1)^2}{2} + \min_{k}\sum_{m=1}^{k}d_{2H}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2 + 0\\ &= \min_k \frac{d_{2H}}{2}((d_{1E}-\Delta_k)^2 + \Delta_{k-1}^2) \quad ,k = 1,2,\dots,z+1. \qquad (24)\end{aligned}$$

CASE II: $d_{2H} \equiv 1 \pmod 2 \Rightarrow$

$$\boldsymbol{\tau} \in S' = \{(1,1,\dots,1),(d_{2H},1,\dots,1),(d_{2H},d_{2H},1,\dots,1),\dots,(d_{2H},\dots,d_{2H})\}$$

and the channel loss is given by

$$\begin{aligned}\gamma(\Lambda_z) &= \sum_{k=1}^{z}\frac{d_{2H}}{2}b(k) + \min_k\{\sum_{m=1}^{k-1}d_{2H}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2 +\\ &\qquad \sum_{m=k}^{z}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2\};\end{aligned} \qquad (25)$$

the first two sums have been calculated above, and it is easy to show using the fact that $\Delta_{z+1} = d_{1E} - \Delta_z$

$$\sum_{m=k}^{z}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1}) = 2\Delta_z^2 - (d_{1E}-\Delta_k)^2 - \Delta_{k-1}^2;$$

$$\text{hence, } \gamma(\Lambda_z) = \min_k \frac{(d_2-1)}{2}((d_{1E}-\Delta_k)^2+\Delta_{k-1}^2)+\Delta_z^2 \quad ,k=1,2,...,z+1. \tag{26}$$

The minimum channel loss in the case of its optimal strategy under the condition of decoding failure is given in (24) and (26).

4. OPTIMAL DECODER STRATEGY

The optimal decoder strategy maximizes the channel loss for a fixed z; i.e., it is the solution of the equation

$$\Lambda_z^* = \arg \max_{\Lambda_z} \gamma(\Lambda_z)$$

with the resulting channel loss for d_{2H} even

$$\gamma = \max_{\Lambda_z} \min_k \frac{d_{2H}}{2}[(d_{1E}-\Delta_k)^2+\Delta_{k-1}^2] \quad k=1,2,...,z+1.$$

Let $\delta_k = \frac{\Delta_k}{d_{1E}}$ and $f(\delta_k,\delta_{k-1}) = (1-\delta_k)^2+\delta_{k-1}^2$. It is shown in the Appendix that maximum in

$$\max_{\Lambda_z} \min_k f(\delta_k,\delta_{k-1})$$

is obtained when $f(\delta_k,\delta_{k-1}) = \alpha_z$, $k=1,2,...,z+1$; where α_z is some unique constant that depends on z. Given the conditions $\delta_0 = 0$ and $\delta_{z+1} = 1-\delta_z$; the $z+1$ equations for $\delta_1,...,\delta_z,\alpha_z$ can be solved numerically. This determines the optimal decoder strategy through $\Delta_i = \delta_i \, d_{1E}$. For d_{2H} odd the channel loss is given by

$$\gamma = \max_{\Lambda_z} \min_k \frac{(d_{2H}-1)}{2}[(d_{1E}-\Delta_k)^2+\Delta_{k-1}^2]+\Delta_z^2 \quad k=1,2,...,z+1.$$

It can be shown that the solution for the above game is the same as the even case. That is, the strategy is the same and the resulting losses are equal, for both cases.

The error-correcting capability (squared) of the code when the decoder uses its best strategy is

$$\gamma = \frac{d_{2H}d_{1E}^2}{2}\alpha_z \quad \text{or} \quad \sqrt{\gamma} = \frac{d_{1E}\sqrt{d_{2H}}}{2}\beta_z; \; \beta_z < 1 \; . \tag{27}$$

Proposition:

$$\lim_{z\to\infty} \beta_z = 1$$

Proof: α_z is given by

$$\alpha_z = (1-\delta_k)^2+\delta_{k-1}^2$$

where δ_k satisfies the conditions:

1. δ_k is a strictly increasing function of k ;
2. δ_k is bounded above by 0.5.

The second property uses the fact that $\alpha_z < 0.5$ (see Appendix), also $\delta_z = \sqrt{\alpha_z/2} < 0.5$. Now assume $\delta_{k+1} < 0.5$, then we have $\delta_k = \sqrt{\alpha_z - (1-\delta_{k+1})^2} < 0.5$; therefore, $\delta_k < 0.5$ for all $k = 1, 2, ..., z$, by induction. The above two facts imply that δ_k converges as $k, z \rightarrow \infty$; moreover,

$$\begin{aligned} \alpha_\infty &= (1-\delta_\infty)^2 + \delta_\infty, \\ \alpha_\infty &= 2\delta_\infty. \end{aligned}$$

Solving these two equations for α_∞ we get

$$\alpha_\infty = 0.5,$$

and the proposition follows. Consequently, the full error correcting capability of the concatenated code is achieved asymptotically. Table 1 shows that over 95 % of the full error correcting capability is realized for $z = 4$, with values: $\delta_1 = 0.317, \delta_2 = 0.395, \delta_3 = 0.443, \delta_4 = 0.481$.

Table 1: Error-Correcting capability of the code given by β_z for various number of branches z.

z	α_z	β_z
2	0.342	0.827
3	0.450	0.948
4	0.466	0.965
5	0.475	0.974
7	0.485	0.984
9	0.490	0.990
14	0.495	0.994
33	0.499	0.999

5. CONCLUSION

The error correcting properties of concatenated codes was evaluated for soft decision (inner) decoding. We developed an algorithm which uses error-erasure decoding and the use of several branches with different tentative decisions (i.e., parallel decoding). This decoding algorithm is optimal for a fixed number of decoders.

The algorithm developed by Zyablov for hard decision decoding is capable of maximum random error correction which is achieved with a finite number of decoders. Although the maximum error correction in our case is only realized asymptotically with the number of decoders, Table 1 shows that only a few number of decoders are needed to achieve over 95 % of the maximum correction capability.

APPENDIX

Let $f(\delta_k, \delta_{k-1}) = (1-\delta_k)^2 + \delta_{k-1}^2$ be a sequence of functions defined for $k = 1, 2, ..., z+1$ with the following properties:

1. $\delta_k \in (0, \frac{1}{2})$ for $k = 1, 2, ..., z$
2. $\delta_0 = 0$
3. $\delta_{z+1} = 1 - \delta_z$

Also, define $\Lambda_z = \{\delta_1, \delta_2, ..., \delta_z\}$. We need to find

$$\max_{\Lambda_z} \min_{k} f(\delta_k, \delta_{k-1}). \tag{28}$$

We will show first that it is necessary to have $f(\delta_k, \delta_{k-1}) = \alpha, k = 1, ..., z+1$, α being some constant, then prove α is unique for a given integer z.

Necessary condition:

Assume $\exists$ some strategy Λ_z such that

$$f(\delta_k, \delta_{k-1}) = \alpha.$$

To prove the necessary condition, we need to show that for any $\Lambda'_z \neq \Lambda_z$ [2]

$$\max_{\Lambda'_z} \min_{k} f(\delta_k, \delta_{k-1}) \; < \; \alpha.$$

For any strategy $\Lambda_z \neq \Lambda'_z$ we have $\delta_j > \delta'_j$ or $\delta_j < \delta'_j$, for at least one j. Now let

$$j = \arg\min_{k} \{\delta_k : \delta_k < \delta'_k\};$$

if such j exists, then it is obvious that $(1-\delta'_j)^2 + \delta'^2_{j-1} < (1-\delta_j)^2 + \delta^2_{j-1} = \alpha$; if such j does not exist, let $l = \arg\max_k\{\delta_k : \delta_k > \delta'_k\}$. Then we have, $(1-\delta'_{l+1})^2 + \delta'^2_l < (1-\delta_{l+1})^2 + \delta^2_l = \alpha$; we know that l or j, or both, exist. Hence, the optimal strategy is Λ_z.

Uniqueness:

Assume there exists strategies Λ and $\Lambda\prime$ such that (for $k = 1, 2, ..., z+1$)

$$f(\delta_k, \delta_{k-1}) = \alpha \;\;,\;\; f(\delta'_k, \delta'_{k-1}) = \alpha', \;\; \alpha < \alpha'.$$

Solving $(1-\delta_k)^2 + \delta^2_{k-1} = \alpha$ for δ_k we get $\delta_k = 1 - \sqrt{\alpha - \delta^2_{k-1}}$. Furthermore, $\delta_1 = 1 - \sqrt{\alpha} > \delta'_1 = 1 - \sqrt{\alpha'}$; assume $\delta_k > \delta'_k$, then $1 - \sqrt{\alpha - \delta^2_k} = \delta_{k+1} > 1 - \sqrt{\alpha' - \delta'^2_k} = \delta'_{k+1}$; hence, by induction, $\delta_k > \delta'_k \;\; \forall k$; in particular, for $k = z$, $\delta_z > \delta'_z \Rightarrow$ [3] $\sqrt{\alpha/2} > \sqrt{\alpha'/2} \Rightarrow \alpha > \alpha'$, which contradicts our assumption that $\alpha < \alpha'$.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under grant ECS-8451266.

---

[2] i.e., $\delta'_k \neq \delta_k$, for some k.

[3] $(1-\delta_{z+1})^2 + \delta^2_z = \alpha$ or $\delta_z = \sqrt{\frac{\alpha}{2}}$.

REFERENCES

[1] G.D. Forney, *Concatenated Codes*, MIT research monograph No. 37, The MIT press, Cambridge, Mass. 1966.

[2] V.V. Zyablov, "Optimization of concatenated decoding algorithms," *Probl. Peredachi Inf.* vol. 9, No.1, pp. 26-32, 1973.

[3] T. Ericson, "Analyses of the Blokh-Zyablov decoding algorithm," Internal Report (1986).

[4] V.V. Zyablov, *Linear Concatented Codes*, Moscow 1982 (in Russian).

[5] A.A. Hassan, W.E. Stark,"On decoding concatenated codes," *IEEE Trans. Inform. Theory*, submitted for publication.

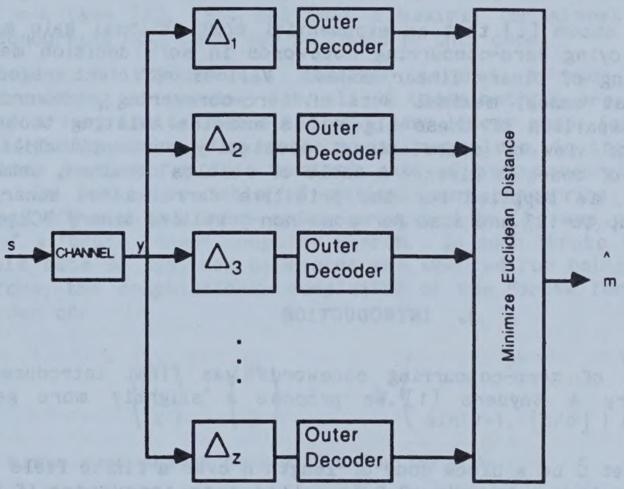

s
CHANNEL
y
Δ_1
Δ_2
Δ_3
Δ_z
Outer Decoder
Outer Decoder
Outer Decoder
Outer Decoder
Minimize Euclidean Distance
$\hat{m}$

Fig. 1. Parallel Concatenated Decoding

EFFICIENT ALGORITHMS FOR THE SEARCH OF ZERO-CONCURRING CODEWORDS

ALEXANDER VARDY and **YAIR BE'ERY**

Department of Electronic Communications Control and Computer Systems,
Tel-Aviv University, Ramat Aviv, Tel-Aviv, Israel.

This work was supported in part by the Foundation for Research in Electronics, Computers and Communications, administered by the Israel Academy of Sciences and Humanities.

ABSTRACT

It has been shown [1] that an exponential computational gain may be obtained by employing zero-concurring codewords in soft decision maximum likelihood decoding of binary linear codes. Various efficient algorithms providing, in most cases, maximal sets of zero-concurring codewords are presented. A comparison of these algorithms and the existing techniques from the point of view of computational complexity and applicability to general families of codes is given. A table of results obtained, using the above algorithms, is supplied for the primitive narrow-sense binary BCH codes of length up to 127 and also for some non-primitive binary BCH codes.

I. INTRODUCTION

The concept of zero-concurring codewords was first introduced and defined by Be'ery & Snyders [1][1].We propose a slightly more general definition.

Definition 1: Let $\tilde{C}$ be a block code of length n over a finite field $GF(q)$. A set of j independent codewords of $\tilde{C}$ is called **zero-concurring** if in all of the n positions at most one of the j codewords has a nonzero component.

Any **two** codewords will be called **non-concurring** if they constitute a zero-concurring set; otherwise they are called **concurring**.

We shall confine ourselves to the consideration of **binary** codes only, although the results presented in this paper may be straightforwardly extended, so as to be applicable to codes over an arbitrary finite field.

It was shown in [1], [2], [3], that a considerable computational gain may be obtained by employing zero-concurring codewords in optimal (maximum likelihood) soft decision decoding of low-rate binary linear codes. This computational gain increases exponentially with the number j of zero-concurring codewords found in the code. For instance, in the case of the [23,12,7] binary Golay code decoding complexity is reduced from 49,152 operations to only 4,032 operations, when incorporating a set of 3 zero-concurring codewords in the decoding procedure (see [1], [2], [3] for other examples).

---

[1] The term 'zero-concurring' stems from the word concurring, encountered in the literature on majority logic decoding [7]. Some of the subcodes of the Golay code discussed by Conway & Sloane [2] and the 'coset codes' of Forney [3] are spanned by sets of zero-concurring codewords, though no specific definition for these codewords was given in either [2] or [3].

Given a binary linear block code C[n,k,d] of length n and dimension k, let J denote the number of elements in a maximal set of zero-concurring codewords of C. Evidently,

$$J \leq \left\lfloor \frac{n}{d} \right\rfloor \tag{1}$$

where d is the minimum Hamming weight of C.

It was shown in [1] that for certain codes the foregoing upper bound is achievable, and it will be demonstrated in this paper that for the predominant majority of many other codes a set of $\left\lfloor \frac{n}{d} \right\rfloor$-1 zero-concurring codewords can be found. This implies that an appreciable reduction of decoding complexity is achievable for any binary linear code, provided that k-J < n-k (see [4], [5]) and that a maximal (or almost maximal) set of zero-concurring codewords can be found in this code.

Be'ery & Snyders [1] consider several methods of constructing zero-concurring codewords. Yet, all of these methods are either applicable only to a very limited sub-class of binary linear codes (such as the family of Reed-Muller codes, for instance) or incomplete, in the sense that they sometimes provide far from maximal sets of zero-concurring codewords, or both. In fact, the table of maximal zero-concurring subsets of some Hamming, BCH and the Golay codes, presented in [1] was compiled with the help of a "brute force" computer search. In such "brute force" search all possible sets of distinct codewords are checked for being zero-concurring. Therefore, the computational complexity of the "brute force" search is of the order of:

$$\binom{2^k}{2} + \binom{2^k}{3} + \ldots + \binom{2^k}{\min\{J+1, \lfloor n/d \rfloor\}} \tag{2}$$

which makes such search prohibitive for large k.

In this paper we derive several efficient algorithms for the search of zero-concurring codewords, that are directly applicable to any binary (linear) code. The computational complexity of all the proposed algorithms is substantially lower than that of the "brute force" approach. In Section II we present our basic recursive algorithm, - Algorithm A. At the end of the section a version of Algorithm A with reduced memory complexity is derived. Algorithm A is complete. Its computational complexity is, however, relatively high. In Section III we propose a more efficient algorithm, - Algorithm B. Maximal zero-concurring subsets have been found by means of Algorithm B in all the codes to which this algorithm had been applied. Nevertheless, it has not been proved that the algorithm is complete. Algorithm C, which is presented in Section IV, has the lowest (polynomial) computational complexity; yet it is incomplete when applied to certain codes.

II. THE BASIC RECURSIVE ALGORITHM

Evidently, a set of distinct nonzero codewords is zero-concurring if and only if any two of its components are non-concurring. Hence, an efficient recursive algorithm, providing a maximal set of zero-concurring codewords can be formulated as follows:

Algorithm A:

Let $\tilde{C}$ be a (non-linear) block code.

1. Select a nonzero codeword $\mathbf{c} \in \tilde{C}$. Let $\tilde{C}'$ be the set of all the codewords of $\tilde{C}$ that are non-concurring to the selected codeword $\mathbf{c}$.

2. If $\tilde{C}' = \{0\}$ or $\tilde{C}' = \phi$, set $A_c = \{\mathbf{c}\}$; otherwise find by means of Algorithm A, a maximal zero-concurring subset of $\tilde{C}'$, say A'_c, and set:

$$A_c = A'_c \cup \{\mathbf{c}\}.$$

3. Repeat steps 1 and 2 for all the nonzero codewords of $\tilde{C}$, unless at some execution of step 2 a bound reaching set of zero-concurring codewords is found.

A set A^*_c, such that:

$$|A^*_c| \geq |A_c| \qquad \text{for all} \qquad \mathbf{c} \in \tilde{C}-\{0\}$$

where $|A|$ stands for the number of elements in the set A, is a maximal zero-concurring subset of the code $\tilde{C}$.

The principal idea of the foregoing algorithm is to consider only zero-concurring subsets of $\tilde{C}$ instead of all the possible subsets of $\tilde{C}$ as in the "brute force" approach. Since the number of zero-concurring subsets is generally much less than the number of all the possible subsets of the code, substantial computational gain follows.

Let M and M' be the number of codewords in $\tilde{C}$ and $\tilde{C}'$, respectively. We assume that steps 1 and 2 of the algorithm are executed exactly M times. At step 1 of the algorithm M comparison operations are performed. Therefore, the computational complexity of Algorithm A can be expressed by the following recursive formula:

$$N_A(M) = M\cdot[M + N_A(M')] = M^2 + M\cdot N_A(M') \qquad (3)$$

If $\tilde{C}$ is a binary linear code C[n,k,d] then, as it will be shown in the sequel:

1) $\tilde{C}'$ is also a binary linear code;
2) M' can be approximated by $\dfrac{M}{2^d}$.

Succesively applying formula (3) we obtain:

$$N_A(M) = M^2 + M\cdot N_A(M') \approx M^2 + M\cdot N_A\left(\frac{M}{2^d}\right) \approx M^2 + M\cdot\left[\frac{M^2}{2^{2d}} + \frac{M}{2^d} N_A\left(\frac{M}{2^{2d}}\right)\right] \approx$$

$$\approx \dots \approx M^2 + \frac{M^3}{2^{2d}} + \frac{M^4}{2^{5d}} + \frac{M^5}{2^{9d}} + \frac{M^6}{2^{14d}} + \dots \qquad (4)$$

Clearly, at most J stages of recursion are at all possible in Algorithm A. Therefore, the series above terminates after at most J terms. Substituting 2^K for M gives the following estimate for the computational complexity of Algorithm A when applied to binary linear codes:

$$N_A(k,d,J) = \sum_{r=0}^{J-1} 2^{[(r+2)k-(\alpha_{r+1}-1)d]} = 2^{2k} + 2^{3k-2d} + 2^{4k-5d} + 2^{5k-9d} + \dots \qquad (5)$$

$$\text{where } \alpha_r = \sum_{i=0}^{r} i,$$

The right hand side of (5) is, indeed, much less than the computational complexity of the "brute force" approach given in (2).

The formulation of our next algorithm demands a certain elaboration of the well-known technique of **shortening** (or taking **cross-section** of) binary linear codes (see [6]).

Let G be the generator matrix of C[n,k,d]; then:

Definition 2: For each codeword $\mathbf{c} \in C$, the cross-section weight of $\mathbf{c}$, - $w^{cs}(\mathbf{c})$, is the rank of the submatrix $\Delta_\mathbf{c}$ of G consisting of those columns that correspond to the nonzero coordinates of $\mathbf{c}$.

The cross-section weight of a codeword does not depend on the choice of G since the rank of a matrix is invariant under elementary row operations. In fact, cross-section weight defines a norm in a linear space C in much the same way as Hamming distance defines a metric in this space.

Theorem 1: The set C' of all the codewords of C that are non-concurring to a specific codeword $\mathbf{c} \in C$ is a binary **linear** code C'[n',k',d'] obtained by means of shortening the original code C by all the nonzero coordinates of $\mathbf{c}$, where:

$$n' = n - w_H(\mathbf{c}) \quad ; \quad k' = k - w^{cs}(\mathbf{c}) \quad ; \quad d' \geq d.$$

and $w_H(\mathbf{c})$ stands for the Hamming weight of $\mathbf{c}$.

Proof: Let $L = \{i_1, i_2, \dots i_\ell\}$, where $\ell = w_H(\mathbf{c})$, be the set of all the nonzero coordinates of $\mathbf{c}$. As the rank of $\Delta_\mathbf{c}$ is by definition $w^{cs}(\mathbf{c})$, there exists a nonsingular k×k matrix E, such that:

$$E\Delta_\mathbf{c} = \begin{bmatrix} & & & \Delta_\mathbf{c}^* & & \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & . & . & . & 0 \\ . & . & & & & . \\ . & . & & & & . \\ 0 & 0 & . & . & . & 0 \end{bmatrix}$$

where $\Delta_\mathbf{c}^*$ is a matrix consisting of $w^{cs}(\mathbf{c})$ independent rows and the number of zero rows at the bottom of $E\Delta_\mathbf{c}$ is $k-w^{cs}(\mathbf{c})$. As the matrix E is nonsingular, EG is also a generator matrix of C. Let G' be a $(k-w^{cs}(\mathbf{c}))\times n$ matrix obtained by deleting the top $w^{cs}(\mathbf{c})$ rows from EG. By construction, G' is a submatrix of EG, such that the columns corresponding to the nonzero coordinates of $\mathbf{c}$ contain only zero entries. Therefore G' is a generator matrix of the shortened code of C given by:

$$C' = \{\mathbf{c} = (c_1, c_2, \dots c_n) \mid \mathbf{c} \in C \text{ and } c_i = 0 \text{ for all } i \in L\} \qquad (6)$$

It follows from (6) that C' is, indeed, the required set of all the codewords of C that are non-concurring to **c**. Since $C' \subset C$, the minimum distance of C' is at least d. The dimension of C' is by construction:

$$k' = \text{rank}\ (G') = k - w^{cs}(\mathbf{c})$$

Evidently, those coordinates of C' that are identically zero may be either ignored or deleted. Therefore, the blocklength of C' is given by:

$$n' = n - |L| = n - w_H(\mathbf{c})$$ **QED**

It follows from the proof of Theorem 1 that the generator matrix G' of the shortened code may be computed by means of performing at most $k \cdot w_H(\mathbf{c})$ elementary row operations on the matrix G. Indeed, G must have at least one nonzero entry in column $i_1 \in L$. By applying k appropriate elementary row operations we obtain a generator matrix E_1G of C having only one nonzero entry in column i_1, this entry being located at the top of the column. We delete the top row of E_1G and proceed in the same manner for all $i \in L$ until the required matrix G' is obtained. We shall refer to this procedure in the sequel as "performing cross-section on G by **c**".

In light of the preceding discussion a different version of the basic recursive algorithm, named Algorithm A^{cs}, may be conceived. Algorithm A^{cs} is similar to Algorithm A, except for the following:

1) it is applicable only to **linear** codes,
2) the set C' is obtained by performing cross-section on the generator matrix of C by the selected codeword **c**.

Reasoning as we did when deriving (3) one can obtain the following recursive expression for the upper bound on the computational complexity of Algorithm A^{cs}:

$$N_{A^{cs}}(M,n,k) \leq \sum_{\mathbf{c} \in C} [k + k \cdot w_H(\mathbf{c}) + N_{A^{cs}}(M',n',k')] \tag{7}$$

where the first term in the brackets stands for k row additions required for the "encoding", i.e. computing a codeword from the generator matrix. Evidently:

$$\sum_{\mathbf{c} \in C} (w_H(\mathbf{c}) + 1) \leq M \cdot \left(\frac{n}{2} + 1 \right) \approx \frac{Mn}{2} \tag{8}$$

It follows from Theorem 1 that:

$$M' = \frac{M}{2^{w^{cs}(\mathbf{c})}} \leq \frac{M}{2^{d^{cs}}} \tag{9}$$

where d^{cs} is the minimum cross-section weight of C. Approximating d^{cs} by d leads to the same estimate of M' as the one we used in the derivation of (4), namely:

$$M' \approx \frac{M}{2^d} \tag{10}$$

Combining (7) with (8), (9), (10) and proceding as in (4) we obtain the following estimate of the computational complexity of Algorithm A^{cs}:

$$N_{A^{cs}}(M,n,k,d,J) \approx \sum_{r=0}^{J-1} \frac{n_r k_r}{2} \cdot \frac{M^{(r+1)}}{2^{\alpha_r d}} \tag{11}$$

where n_r and k_r are the average blocklength and dimension of a code obtained at stage r of recursion in Algorithm A^{cs}. Substituting 2^k for M gives:

$$N_{A^{cs}}(n,k,d,J) \approx \sum_{r=0}^{J-1} \frac{n_r k_r}{2} \cdot 2^{[(r+1)k-\alpha_r d]} \leq \frac{nk}{2} \cdot \sum_{r=0}^{J-1} 2^{[(r+1)k-\alpha_r d]} =$$

$$= \frac{nk}{2} \cdot [2^k + 2^{2k-d} + 2^{3k-3d} + 2^{4k-6d} + 2^{5k-10d} + \dots] \tag{12}$$

Apparently, the right hand side of (12) is less than (5) whenever:

$$d \geq \log_2(nk). \tag{13}$$

Another important feature of the cross-section version of Algorithm A is **low memory complexity.** Memory complexity of Algorithm A grows exponentially with k since the entire code must be stored in memory. Memory complexity of Algorithm A^{cs}, however, increases linearly with k as all the operations are performed on the rows of a generator matrix.

III. COMPUTATIONAL GAIN BY EMPLOYING A SELECTION CRITERION

Computational complexity of Algorithm A may be considerably reduced by introducing a **selection criterion,** that could be used to distinguish between the elements of a **maximal** zero-concurring subset of the code and all other codewords of C. This removes the necessity to repeat steps 1 and 2 of the algorithm for all the nonzero codewords of C and, thus, results in a substantial computational gain.

Let A_c be one of the 2^k-1 zero-concurring subsets of C, generated by Algorithm A^{cs}, and let A_c^* be a maximal zero-concurring subset of C, eventually resulting from the algorithm. We define the **average cross-section weight** of a set A_c as:

$$W^{cs}[A_c] = \frac{1}{|A_c|} \sum_{c_r \in A_c} w_r^{cs}(c_r) \tag{14}$$

where $w_r^{cs}(c_r)$ is the cross-section weight of a codeword $c_r \in A_c$ in the code obtained at stage r of recursion in Algorithm A^{cs}.

By Theorem 1:

$$\sum_{c_r \in A_c} w_r^{cs}(c_r) = \sum_{c_r \in A_c^*} w_r^{cs}(c_r) = k \tag{15}$$

and since $|A_c^*| \geq |A_c|$,

$$W^{cs}[A_c^*] \leq W^{cs}[A_c] \tag{16}$$

It follows from (16) that if one of the 2^k-1 sets A_c could be chosen, such that its average cross-section weight would be the smallest, - this set would be a maximal zero-concurring subset of C.

Hence we suggest the **minimum cross-section weight** selection criterion to be employed at step 1 of the following algorithm.

Algorithm B:

Let G be a generator matrix of a binary linear code C[n,k,d].

1. Select a minimum cross-section weight codeword of C, say **v**, and obtain the generator matrix G' of a C'[n',k',d'] code by performing cross-section on G by **v**.
2. If $k' = 0$ set $A_{\mathbf{v}} = \{\mathbf{v}\}$; otherwise find, by means of Algorithm B, a zero-concurring subset of C', say $A'_{\mathbf{v}}$, and set:

$$A_{\mathbf{v}} = A'_{\mathbf{v}} \cup \{\mathbf{v}\}$$

It is conjectured that the set $A_{\mathbf{v}}$ is a **maximal** zero-concurring subset of C. In fact, no example to the contrary has been hitherto found.

We assume that a minimum cross-section weight codeword of C is selected with the help of a complete search through the code and that it takes on the average $\frac{nk}{2}$ elementary row operations to determine the cross-section weight of a codeword. Then the computational complexity of Algorithm B is given by:

$$N_B(n,k) = \frac{1}{2}\cdot[nk2^k+n_1(k-d_0^{cs})2^{k-d_0^{cs}}+n_2(k-d_0^{cs}-d_1^{cs})2^{k-d_0^{cs}-d_1^{cs}}+\dots] \approx \frac{nk}{2}\cdot 2^k \quad (17)$$

where d_r^{cs} and n_r are the minimum cross-section weight and the blocklength of the code obtained at stage r of recursion in Algorithm B.

VI. A POLYNOMIAL TIME ALGORITHM

Clearly, the computational complexity of Algorithm B is much lower than either (5) or (12); nevertheless it still grows exponentially with k. A **polynomial time** (incomplete) algorithm will be derived with the help of the following result.

Theorem 2: Let $C^{\perp}[n,n-k,d^{\perp}]$ be the dual code of C[n,k,d]. Then the minimum cross-section weight of C is bounded by:

1) $d^{cs} \leq k$,
2) $d^{cs} \leq d$,
3) $d^{cs} \geq \min\,[d^{\perp}-1,d]$

Proof: 1) $\forall\, \mathbf{c} \in C,\ \ w^{cs}(\mathbf{c}) = \text{rank}\,(\Delta_{\mathbf{c}}) \leq \text{rank}\,(G) = k.$

2) Let $\mathbf{u} \in C$ be a codeword of Hamming weight d. Then:

$$w^{cs}(\mathbf{u}) = \text{rank}\,(\Delta_{\mathbf{u}}) \leq d$$

as $\Delta_{\mathbf{u}}$ contains only d columns.

3) Since G is a parity check matrix of $C^{\perp}$ every $d^{\perp}-1$ columns of G are linearly independent. Therefore, if $d \ge d^{\perp}-1$ we may write down:

$$\forall\, \mathbf{c} \in C,\ w^{cs}(\mathbf{c}) \ge d^{\perp}-1$$

If $d < d^{\perp}$, codewords $\mathbf{c} \in C$ exist such that $w_H(\mathbf{c}) \le d^{\perp}-1$. For these codewords $\Delta_{\mathbf{c}}$ contains $w_H(\mathbf{c})$ linearly independent columns, so that:

$$w^{cs}(\mathbf{c}) = \text{rank}\,(\Delta_{\mathbf{c}}) = w_H(\mathbf{c}) \ge d \qquad \textbf{QED}$$

It follows from Theorem 2 and its proof that whenever $d < d^{\perp}$, $d^{cs} = d$ and a minimum Hamming weight codeword of the code is also a minimum cross-section weight codeword. This justifies, to some extent, the substitution of the minimum cross-section weight selection criterion by the **minimum Hamming weight selection criterion**, which in turn leads to the algorithm named **Algorithm C.**

Algorithm C is not complete, in the sense that it does not necessarily provide a maximal set of zero-concurring codewords. Consider the [12,4,3] binary code with the following generator matrix:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

It can be readily verified that the maximal zero-concurring subset of the code is:

$$\begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

Yet, applying Algorithm C to the code gives:

$$\begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Two more examples are the BCH [63,18,21] code, where a set of J=3 zero-concurring codewords have been found by means of Algorithm B, and the BCH [127,78,15] code, where a set of J=8 zero-concurring codewords have been constructed by the methods of Be'ery & Snyders [1]. Application of Algorithm C to these codes provides zero-concurring subsets with only J-1 elements.

Apparently, other such codes exist. However, in all but one of the 44 different codes presented in Table I, zero-concurring subsets with $\lfloor n/d \rfloor -1$ or more elements have been found with the help of Algorithm C, where $\lfloor n/d \rfloor$ is but an upper bound as in (1). This experimental evidence implies that Algorithm C may be claimed to be "almost complete", which is not true for the construction methods of [1].

The computational complexity of Algorithm C is given by:

$$N_C(n,k,d) = \sum_{r} N_D(n_r,k_r,d_r) \tag{18}$$

where r is upper bounded by $\lfloor n/d \rfloor$; n_r, k_r and d_r are the length, the dimension and the minimum distance of the code obtained at stage r of recursion in Algorithm C. $N_D(n_r,k_r,d_r)$ is the computational complexity of the algorithm employed for the search of a minimum Hamming weight codeword in the code. Evidently, if one would use a polynomial time algorithm to find minimum weight codewords (such as the algorithm proposed by Harari in [8]), the computational complexity of Algorithm C would be **polynomial.**

TABLE I

Code	Upper bound	Number of zero-concurring codewords found using the proposed algorithms	Reduction of decoding complexity
Primitive Narrow-Sense BCH			
1. BCH (7,4,3)	2	2	2
2. BCH (15,11,3)	5	5	
3. BCH (15,7,5)	3	3	8
4. BCH (15,5,7)	2	2	4
5. BCH (31,26,3)	10	10	
6. BCH (31,21,5)	6	5	
7. BCH (31,16,7)	4	4	8
8. BCH (31,11,11)	2	2	4
9. BCH (31,6,15)	2	2	4
10. BCH (63,57,3)	21	20	
11. BCH (63,51,5)	12	12	
12. BCH (63,45,7)	9	8	
13. BCH (63,39,9)	7	6	
14. BCH (63,36,11)	5	5	
15. BCH (63,30,13)	4	4	16
16. BCH (63,24,15)	4	4	16
17. BCH (63,18,21)	3	3	8
18. BCH (63,16,23)	2	2	4
19. BCH (63,10,27)	2	2	4
20. BCH (63,7,31)	2	2	4
21. BCH (127,120,3)	42	41	
22. BCH (127,113,5)	25	24	
23. BCH (127,106,7)	18	17	
24. BCH (127,99,9) [see (1)]	14	12	
25. BCH (127,92,11)	11	10	
26. BCH (127,85,13)	9	8	
27. BCH (127,78,15)	8	7	
28. BCH (127,71,19)	6	5	
29. BCH (127,64,21)	6	5	16
30. BCH (127,50,27)	4	3	8
31. BCH (127,43,31)	4	3	8
32. BCH (127,36,31)	4	4	16
33. BCH (127,29,43)	2	2	4
34. BCH (127,22,47)	2	2	4
35. BCH (127,15,55)	2	2	4
36. BCH (127,8,63)	2	2	4

Non-Primitive Narrow-Sense BCH

37. BCH (17,9,5)	3	3	4
38. BCH (21,12,5)	4	3	
39. BCH (23,12,7)	3	3	4
40. BCH (33,22,6)	5	4	
41. BCH (41,21,9)	4	4	8
42. BCH (47,24,11)	4	4	8
43. BCH (65,53,5)	13	12	
44. BCH (65,40,10)	6	5	

(1) The only known instance, where the number of zero-concurring codewords found using Algorithm C is less than $\lfloor n/d \rfloor -1$.

ACKNOWLEDGEMENT

The authors are indebted to Jakov Snyders for constructive discussions. Alexander Vardy wishes to thank Hagit Itzkovich for her invaluable help.

REFERENCES

[1] Y. Be'ery and J. Snyders. "Optimal soft decision block decoders based on Fast Hadamard Transform". IEEE Trans. Inform. Theory, vol. IT-32, pp. 355-364, 1986.

[2] J.H. Conway and N.J.A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice; IEEE Trans. Inform. Theory, vol. IT-32, pp. 41-50, 1986.

[3] G.D. Forney, Jr., "Coset codes", IEEE Trans. Inform. Theory, to appear.

[4] J.K. Wolf, "Efficient maximum likelihood decoding of linear block codes", IEEE Trans. Inform. Theory, vol. IT-22, pp. 514-517, 1976.

[5] Y. Be'ery and J. Snyders, "A recursive Hadamard Transform optimal soft decision decoding algorithm", to appear in SIAM J. on Algebraic and Discrete Methods, vol. 8, No.4, 1987. Presented in part at the IEEE 1985 Int. Symp. Information Theory, Brighton, England.

[6] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error Correcting Codes, Amsterdam, The Netherlands: North-Holland, 1977.

[7] R.E. Blahut, Theory and Practice of Error Control Codes. Reading, MA: Addison-Wesley, 1983, p. 392.

[8] S. Harari, "A polynomial time algorithm for finding minimum weight codewords in a linear code." presented at the IEEE 1985 Int. Symp. Information Theory, Brighton, England.

CYCLIC CODES AND THEIR WEIGHTS

VERA PLESS
Mathematics Department
University of Illinois - Chicago
Chicago, IL 60680

This paper is concerned with cyclic self-orthogonal codes and what this implies about the weights of code-words.

If e is the generating idempotent of a cyclic code C, we say $C = \langle e \rangle$. Let n be the length of C, a code over $GF(p)$ with $(n,p)=1$. Then the one-dimensional code generated by the all-one vector h is $\langle \frac{1}{n} h \rangle$. Let $E = \langle \frac{1}{n} h \rangle^{\perp}$ and call a code even-like if it is contained in E. A vector $v=(v_0,\cdots,v_{n-1})$ is called even-like if $v \in E$ and in this situation $\sum_{i=0}^{n-1} v_i = 0$ in $GF(p)$. A vector which is not even-like is called odd-like and a code which contains at least one odd-like vector is called odd-like.

Fact 1 [2]. C is odd-like iff h is in C.

We let μ_{-1} denote the coordinate permutation $i \longrightarrow -i \pmod{n}$.

Fact 2 [2]. If $C = \langle e \rangle$, $C^{\perp} = \langle 1 - \mu_{-1}(e) \rangle$.

Theorem 1. If C is a self-orthogonal cyclic code, then C is even-like.

Proof: As $(n,p)=1$, h is not a self-orthogonal vector so that h is not in C. Hence C is even-like.

Corollary. A self-orthogonal binary cyclic code has minimum weight $\geqq 4$ and a self-orthogonal ternary cyclic code has minimum weight $\geqq 6$.

Theorem 2. Let $C = C_1 \perp \langle \frac{1}{n} h \rangle$ where C_1 is a self-orthogonal cyclic code. Let d_0 be the minimum weight of an odd-like vector in C. Then $d_0^2 - d_0 + 1 \geqq n$.

If equality holds, then d_0 is the minimum weight of C and the supports of the vectors of weight d_0 constitute a projective plane of order d_0+1. Further, the vectors of weight d_0 are multiples of vectors all of whose components are 0 or 1.

The next theorem follows easily from facts in [1].

Theorem 3: If $\overline{C}$ is an extended cyclic binary self-dual code of length $n+1$, then all even weight vectors in C have weight $\equiv 0 \pmod 4$ and all odd weight vectors have weight $\equiv n \pmod 4$.

It is interesting to know when cyclic self-orthogonal codes exist and the next theorems address this.

Theorem 4. If $-1=p^i \pmod n$, there are no cyclic self-orthogonal codes of length n over GF(p).

Proof: Suppose $C=\langle e\rangle$ is self-orthogonal. Then $C \subset C^{\perp} = \langle 1-\mu_{-1}(e)\rangle$. As $e(1-\mu_{-1}(e))=e$, $e\mu_{-1}(e)=0$. But $\mu_{-1}(e)=e$ since $-1=p^i \pmod n$. Hence $e=0$.

Theorem 5 [2]. A cyclic self-orthogonal code of length n and dim $\frac{n-1}{2}$ over GF(p) exists iff the order of p (mod n) is odd.

Corollary [1]. A binary, cyclic self-orthogonal code of length n and dim $\frac{n-1}{2}$ exists iff the factorization of n into prime powers, $n = \Pi p_i^{a_i}$ has the properties that each $p_i \equiv \pm 1 \pmod 8$ and if $p_i \equiv 1 \pmod 8$, the order of 2 (mod p_i) is odd.

References

1. V. Pless, J. M. Masley, and J. S. Leon, "On Weights in Duadic Codes", Jrnl. of Comb. Th.(A), Jan. 1987.
2. V. Pless, "Duadic Codes Revisited", preprint.

PERFORMANCE OF HYBRID ARQ CODING FORMATS OVER A BINARY SYMMETRIC CHANNEL

G. M. HUFFMAN and F. D. GARBER
ElectroScience Laboratory
Department of Electrical Engineering
The Ohio State University, Columbus, Ohio 43210

ABSTRACT

This paper is concerned with the design and performance evaluation of error control coding techniques that make effective use of the repeat request feedback links available in Packet Radio systems with ARQ. Using the binary symmetric channel with a noiseless feedback link in stop and wait mode as a model, we analyze the performance of pure ARQ, Type I Hybrid ARQ [1], and Type II Hybrid ARQ [2]. In addition, we introduce and analyze a technique similar to Type II hybrid ARQ that allows flexibility in employing multiple parity retransmissions to improve performance.

The performance criteria used in the evaluation of these techniques are the error probabilities, the throughput, defined as the reciprocal of the expected number of transmissions required for a block of data to be successful; and the *effective coding rate*, defined as the ratio of the number of information bits successfully decoded, to the total number of bits transmitted (including all retransmissions required for a block to be successful).

Pure ARQ is characterized by the transmission of a high-rate codeword where the decoder is used exclusively for error detection. If codeword errors are detected, a "NAK" is sent to the transmitter and the same codeword is retransmitted. Type I Hybrid ARQ is similar to pure ARQ except that the decoder is used to correct as well as detect codeword errors [1]. If a received codeword contains correctable errors, the errors are corrected and reception is acknowledged. If, on the other hand, uncorrectable errors are detected, the receiver requests a retransmission of the same codeword.

Type II Hybrid ARQ, represents an iterative means of data transfer where two codes, $\mathcal{C}_0$ and $\mathcal{C}_1$ are used [2]. The first code is a high-rate (n, k) code used for error detection only. The second code is a rate $r = 1/2$ invertible $(2k, k)$ code used for both for error correction and error detection. Initially the block of k information bits is encoded with $\mathcal{C}_0$ and transmitted. If errors are detected, a "NAK" is sent and the k information bits are encoded to form a $2k$–bit codeword from $\mathcal{C}_1$. The k parity bits resulting from this encoding are then encoded using $\mathcal{C}_0$ and the codeword is transmitted.

If no errors are detected for this second codeword, the information bits are recovered from the decoded k parity bits by an inversion process [2]. Otherwise, a rate $r = 1/2$ codeword is formed from the k information bits from the first received codeword and the k "parity" bits from the second received codeword. If the codeword so constructed contains correctable errors, the errors are corrected. If uncorrectable errors are detected, the received k parity bits are saved in a buffer and a retransmission of the original k information bits (again encoded using $\mathcal{C}_0$) is requested. This process is repeated, with alternate encodings of the k information bits and the k parity bits, until the received information is successfully decoded.

The multiple parity retransmission scheme we consider utilizes a set of m systematic codes $\mathcal{C}_1, \ldots, \mathcal{C}_m$ where the rate of each successive code is decreasing. This scheme is similar to the incremental redundancy scheme based on punctured codes in [3]. The process begins by transmitting a codeword obtained from an encoding of the k information bits using $\mathcal{C}_1$ which is designed for error detection while possibly allowing a small amount of error correction.

If uncorrectable errors are detected, a NAK is sent and a new codeword is formed by encoding the original k information bits using $\mathcal{C}_2$. The $k_2 \leq k$ parity bits resulting from this encoding are then transmitted. The receiver combines the k information bits received from the first transmission, along with the with the k_2 "parity" bits obtained from the second transmission to form a codeword corresponding to a code with lower rate and having the same capability for error detection, and a greater capability for error correction than $\mathcal{C}_1$.

If a correctable number of errors are detected in the composite codeword, the detected errors are corrected and the process is complete. Otherwise, a NAK is sent, a new codeword is formed by encoding the original k information bits using $\mathcal{C}_3$ and $k_3 \leq k_2$ parity bits resulting from this third encoding are transmitted and so on. If, after all m encodings have been performed, the composite codeword is not correctable, the entire m-step process is re-initiated.

By comparatively evaluating the performance of the techniques discussed above, we show [4] that while Pure ARQ provides high reliability, a large number of retransmissions may be required so that the throughput may be low. Type I Hybrid ARQ sustains a higher throughput than pure ARQ, by virtue of the forward error correction capability but a lower rate code must be used. Next, we confirm that Type II Hybrid ARQ yields higher throughput than pure ARQ without reducing the code rate. Finally, we show that by adding the flexibility of using successively longer, lower rate codes for the multiple transmissions, the multiple retransmission scheme can realize a performance improvement in terms of the *effective coding rate* relative to both Type I and Type II hybrid ARQ.

REFERENCES

[1] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs: Prentice-Hall, 1983.

[2] S. Lin and P. S. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Transactions on Communications*, vol. COM–30, no. 7, pp. 1701–1719, August 1982.

[3] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Transactions on Information Theory*, vol. IT–20, no. 3, pp. 388–389, May 1974.

[4] G. M. Huffman, "Performance of hybrid ARQ formats over a binary symmetric channel," M.S. Thesis, The Ohio State University, August 1987.

alternate encodings of the k information bits and the k parity bits until the received information is successfully decoded.

The multiple parity retransmission scheme we consider uses a set of m systematic codes $C_1, \ldots, C_m$ where the rate of each successive code is decreasing. This scheme is similar to the incremental redundancy scheme based on punctured codes in [3]. The process begins by transmitting a codeword obtained from an encoding of the k information bits using C_1 which is designed for error detection while possibly allowing a small amount of error correction.

If uncorrectable errors are detected, a NAK is sent and a new codeword is formed by encoding the original k information bits using C_2. The $n_2 - k$ parity bits resulting from this encoding are then transmitted. The receiver combines the k information bits received from the first transmission along with the $n_2 - k$ parity bits from the second transmission to form a codeword corresponding to a code with lower rate and having the same capability for error detection, and a greater capability for error correction than C_1.

If a correctable number of errors are detected for the combined codeword, the detected errors are corrected and the process is complete. Otherwise, a NAK is sent and a codeword is formed by encoding the original k information bits using C_3, and the parity bits resulting from this third encoding are transmitted and so on. If, after all m encodings have been performed, the complete codeword is not correctable, the entire m-step process is re-initiated.

By comparatively evaluating the performance of the techniques discussed above, we show [4] that while Pure ARQ provides high reliability, a large number of retransmissions may be required so that the throughput may be low. Type I hybrid ARQ achieves a higher throughput than pure ARQ, by virtue of the forward-error-correction capability, but a lower rate code must be used. We confirm that Type II hybrid ARQ yields higher throughput than pure ARQ without reducing the code rate. Finally, we show that by adding the encoding overhead incrementally, using successively longer, lower rate codes for the multiple transmissions, the multiple retransmission scheme can realize a performance improvement in terms of the system throughput over both Type I and Type II hybrid ARQ.

REFERENCES

[1] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs: Prentice-Hall, 1983.

[2] S. Lin and P. S. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Transactions on Communications*, vol. COM-30, no. 7, pp. 1701-1719, August 1982.

[3] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Transactions on Information Theory*, vol. IT-20, no. 3, pp. 388-389, May 1974.

[4] C. M. [illegible], "Performance of hybrid ARQ schemes over a binary symmetric channel," M.S. Thesis, The Ohio State University, August 1987.

INDEX OF AUTHORS

Aalo, V. 155
Aazhang, B. 135
Abbaszadeh, A. D. 598
Abrahamson, K. 624
Acar, L. 1162
Ait-Kheddache, A. 473
Akiyama, K. 1069
Albertos, P. 795
1202
Allebach, J. P. 483
Al-Muthairi, N. F. 558
Anantharam, V. 684
Antsaklis, P. J. 546
Arce, G. R. 513
753
Armoni, Y. 1081
Azadegan, F. 965
Aziz, G. 121
Babendreier, J. 560
Bai, S. 801
Bailey, D. A. 238
1234
Barmish, B. R. 117
Bar-Ness, Y. 928
Basu, S. 217
Be'ery, Y. 610
Bellow, M. E. 64
Berger, T. 919
Berman, N. 721
Bernhard, P. J. 1027
Bertossi, A. A. 46
Bhattacharya, P. P. 998
Bingulac, S. P. 558
Birkemeier, W. P. 453
Block, H. 662
Boles, W. W. 763
Bose, T. 192
Bouthellier, P. R. 1172
Bovik, A. C. 463
773
Bovopoulos, A. D. 979
Bredeson, J. G. 1103
Brown, D. P. 192
Bryant, R. L. 397
Casavant, T. L. 258
Cassandras, C. G. 287
Chang, S-M. 1040
Chiasson, J. 107
Choi, H-A. 4
268
Choi, M-R. 1142
Choi, S. 928
Choudhry, M. A. 562
Chow, B. S. 453
Ciliz, M. K. 203
Cobb, J. D. 544
Colgate, J. E. 821
Comeau, M. A. 634
Conn, H. E 898
Costello, Jr., D. J. 584
Csiszar, I. 917
Cuny, J. E. 238
1234
Dadoun, N. 624
Dandapani, R. 1059
Dasgupta, S. 1133
DeCarlo, R. A. 556
813
Dehne, F. 660
Delp, E. J. 731
775
DeMarco, C. L. 72
De Pano, N. A. 869
Desoer, C. A. 522
Dickinson, B. W 409
Douligeris, C. 989
Dunham, J. G. 967
Efstratiadis, S. N. 493
Emre, E. 534
Enab, Y-M. 839
Ephremides, A. 969
998
Esfahanian, A-H. 34
268
Esfandiari, F. 97
Farrell, J. A. 1123
Farvardin, N. 938
Fellows, M. R. 397
Ferreira, H. C. 596
Fiduccia, C. M. 201
Fine, T. L. 123
Fineberg, S. A. 258
Fischer, T. R. 955
Foias, C. 205

Foster, R. E. 753
Fowler, D. R. 889
Frazho, A. E. 205
Frederickson, G. 879
Garber, F. D. 622
Gburzynski, P. 908
Ge, Q. W. 305
Gelfand, S. 700
Gelfand, S. B. 731
Ghez, S. 1089
Ghosh, B. K. 1172
Gibson, J. D. 955
Gonzalez, O. R. 546
Gonzalez, T. F. 377
387
Goodrich, M. T. 849
Gudmundsson, G. 375
Gunther, O. 859
Gupta, G. 1050
Hached, M. 841
Haddad, A. H. 426
Hagerman, J. W. 238
Hakimi, S. L. 34
373
Hamdan, A. M. A. 560
Halverson, D. R. 163
Hansen, E. R. 54
Harrison, D. D. 938
Hassan, A. A. 600
Hedge, M. V. 564
Heegard, C. 946
Herro, M. A. 594
Ho, J. 121
Ho, Y-C. 279
Hogan, N. 821
Hopkins, M. A. 399
Hough, A. A. 238
Huffman, G. M. 622
Iftar, A. 1152
Ingram, M. A. 426
Iyengar, S. 1059
JaJa, J. 355
Javidi, B. 1142
Jha, N. K. 1050
Kabuli, M. G. 522
Kanefsky, M. 763
Kappos, E. 707
Karatzoglu, M. A. 969
Katsaggelos, A. K. 493
741
Ke, Y. 869
Keil, J. M. 889
Kent, R. E. 1204
Kerrick, D. D. 463
Khalil, H. K. 97
Khargonekar, P. P. 88
Kim, C. 1224
Kim, Y. K. 574
Kinnersley, N. G. 397
Kirkpatrick, D. G. 624
Knight, J. F. 335
Kogan, J. 1200
Kohli, R. 1101
Koo, B. 955
Krishna, H. 203
Krishnaprasad, S. 227
Krogh, B. H. 317
Kumar, S. P. R. 694
Kuo, S. M. 215
Kurki-Gowdara, S. 377
Lai, T. H. 1007
Lakshmanan, K. B. 634
Langston, M. A. 397
Lazar, A. A. 182
248
979
Lee, C. C. 574
Lee, G. 236
Lee, Y.-H. 743
Leung, J. Y-T. 650
Levy, H. 664
Li, J-H. 1113
Li, S. 409
Lindner, D. K. 560
Lodi, E. 46
Lovell, G. H. 829
Lu, C-C. 967
Lu, D. 125
Luccio, F. 46
Luh, J. Y. S. 839
Madani-Esfahani, S. M. 841
Mahajan, S. 14
Mahtre, K. 217
Manacher, G. K. 24
Mankus, T. A. 24
Martel, C. U. 1079
Masuda, S. 345
Mazumdar, R. 989
McHugh, J. 36
McLeish, M. 908
Mendel, J. M. 417
Meybodi, M. R. 1017

Meyer, G. G. L. 1190
Michel, A. N. 1113
1123
Milito, R. A. 664
Mingo, A. C. 1101
Mitter, S. K. 700
Mobed, M. 436
Modestino, J. W. 938
965
Monderer, B. 182
Morant, F. 795
Munro, J. I. 54
Murata, T. 270
Murray, J. J. 829
Naaman, L. 773
Naclerio, N. J. 345
Nakajima, K. 4
345
Narayan, P. 917
Nel, A. L. 596
Nielsen, P. A. 172
Ni, L. M. 268
Ntafos, S. 375
Onaga, K. 305
Ono, N. 305
Oommen, B. J. 54
O'Rourke, J. 869
898
Orsak, G. 135
Oruc, A. Y. 1037
Ozguner, U. 1152
1162
Pagli, L. 46
Panwar, S. R. 1081
Park, S. Y. 743
Passino, K. M. 335
Perl, Y. 36
Peters, J. G. 14
Petersen, I. R. 88
Peyravi, M. H. 1017
Pless, V. 620
Podrazik, L. J. 1190
Popoli, R. F. 417
Porod, W. 1113
1123
Pourboghrat, F. 811
Przytycka, T. 624
Rahgozar, M. 483
Rajala, S. A. 473
Rao, T. S. 513
Reddy, S. M. 1059
Rego, V. 66
1099
Riberio do Val, J. B. 713
Rim, C. S. 4
345
Robertazzi, T. G. 199
248
Rodger, S. 879
Rodriguez, M. A. 215
Rohlicek, J. R. 674
Root, W. L. 436
1181
Rosenkrantz, D. J. 1027
Rosenthal, A. 641
Rosier, L. E. 1214
Rouanne, M. 584
Rushforth, C. K. 598
Sadowsky, J. S. 145
Sadrolhefazi, A. 123
Sain, M. K. 524
Salam, F. M. A. 801
1142
Saluja, K. K. 1069
Scheuermann, P. 641
Schneider, S. 1037
Schwartz, J. T. 843
Schwartz, S. C. 199
1089
Schwederski, T. 258
Seo, J. H. 534
Serakos, D. 1181
Sharir, M. 843
Sherlekar, D. D. 355
Sherman, P. J. 205
Shi, P. 503
Shrivastava, Y. 1133
Shwartz, A. 721
Silva, M. 270
Simaan, M. 763
785
Sims, C. S. 562
Sira-Ramirez, H. 78
Song, J. 775
Sreenivas, R. S. 315
Stark, W. E. 564
600
Stojemnovic, I. 660
Strickland, S. G. 287
Sullivan, B. J. 741
Sung, Y. Y. 365
Szpankowski, W. 66
1099

Tai, H.-M. 534
Tan, H. L. 731
Tanida, T. 305
Telang, V. 594
Teneketzis, D. 564
Thistle, J. G. 327
Thomas, J. 172
Thomopoulos, S. C. A. 155
Thompson, M. W. 163
Thulasiraman, K. B. 634
Tornero, J. 1202
Towsley, D. 1081
Tuan, T-C. 373
Vakili, P. 279
Vanderwiele, J. M. 1103
VanLandingham, H. F. 399
Vastola, K. S. 1101
Vayada, T. P. 1079
Vardy, A. 610
Verdu, S. 1089
Viniotis, I. 998
Viswanathan, R. 155
Wang, Y. 1142
Ward, R. K. 503
Wessel, A. E. 446
Wey, C-L. 1040
Wicks, M. A. 556
Willsky, A. S. 674
Wise, G. L. 446
Wolf, J. K. 1081
Wong, E. 859
Wonham, W. M. 1
327
Wright, D. A. 596
Wyman, B. F. 524
Yang, T. H. 1007
Yao, K. 125
Yaz, E. 119
Yen, H.-C. 1214
Yeung, R. 919
Young, G. H. 650
Yousef, H. 785
Zak, S. H. 813
841
Zazanis, M. A. 297
Zheng, S-Q. 387
Zhou, K. 88
Zhu, Y. 409
Zribi, M. 107

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C001
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1987:1
3 0112 008149228